

Japanese Patent Laid-open Publication No. HEI 5-134734 A

Publication date : June 1, 1993

Applicant : Mitsubishi Denki K. K.

Title : NUMERICAL VALUE CONTROL UNIT AND CONTROL METHOD THEREFOR

5

[Abstract]

[Object] The present invention is for improving the data input/output performance of a numerical value control unit, and this facilitates a high-speed data input processing, and makes it possible to input/output data in a character code.

[Structure] A character code conversion processing section 31 is provided in a data input/output control section 30, and data input/output is carried out in a character code between the numerical value control unit and an external input/output device 2.

[Effects] It becomes easy to carry out input/output, and it becomes easy to process the output data.

[0228] The input of compile data is for sequentially reading one block data from a memory area that stores a result of an analysis of a work program, as shown in Fig. 58. Data stored in an internal memory (15 or 16) of the NC unit 1 or an external memory unit 47 is read out. In the CAM system 48, if it is possible to generate data similar to the analysis result according to the present invention, it is also possible

to read the data directly from the input/output control unit
22 via the external input/output unit 2. If it is possible
to use the data generated in the CAM system 48, it is also
of course possible to temporarily store this data in the external
5 memory unit 47, and then read this data when necessary.

[Brief Description of the Drawings]

[Fig. 51] This is a key-portion block diagram showing the
10 outline of a work program processing of the NC unit according
to the present invention.

[Explanation of Reference Symbols]

1. NC unit
- 15 2. external input/output unit
22. input/output control unit
23. work program analyzing section
24. data buffer
25. machine control section
- 20 26. servo control section
27. servo motor
31. character code conversion processing section
47. external memory unit
48. CAM system

(19)日本国特許庁(JP)

(12)公開特許公報(A)

(11)特許出願公開番号

特開平5-134734

(43)公開日 平成5年(1993)6月1日

(51)Int.Cl.⁵

G 0 5 B 19/403
19/405

識別記号

庁内整理番号

X 9064-3H

K 9064-3H

C 9064-3H

F I

技術表示箇所

審査請求 未請求 請求項の数30(全 62 頁)

(21)出願番号 特願平4-62143

(22)出願日 平成4年(1992)3月18日

(31)優先権主張番号 実願平3-74367

(32)優先日 平3(1991)9月17日

(33)優先権主張国 日本(JP)

(71)出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72)発明者 丹羽 友光

名古屋市東区矢田南五丁目1番14号 三菱

電機株式会社名古屋製作所内

(74)代理人 弁理士 高田 守

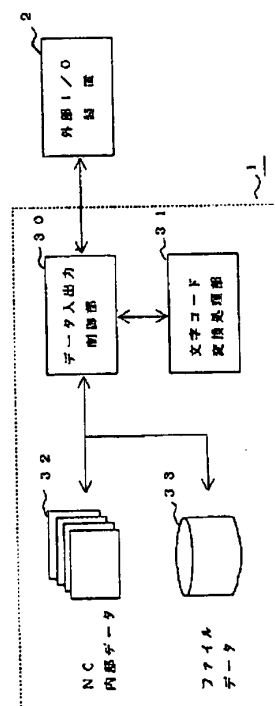
(54)【発明の名称】 数値制御装置及びその制御方法

(57)【要約】

【目的】 本発明は、数値制御装置のデータ入出力機能を向上させるものであり、データの入力処理を高速かつ容易なものとし、データを文字コードで入出力可能とするものである。

【構成】 データの入出力制御部30に文字コード変換処理部31を設け、数値制御装置と外部の入出力機器2との間で文字コードによってデータの入出力を行う構成とする。

【効果】 入出力が容易に行え、出力されたデータが容易に処理可能となる。



【特許請求の範囲】

【請求項1】 数値制御データを格納するメモリを備え、このメモリに格納されるデータを、データ部と、少なくともこのデータ部の表示画面番号及びデータの画面上的表示位置を特定する情報を有するヘッダ情報部とから構成したことを特徴とする数値制御装置。

【請求項2】 請求項1に記載の数値制御装置において、上記データを外部入出力装置との間で、文字コードで入出力させる文字コード変換部を設けたことを特徴とする数値制御装置。

【請求項3】 請求項2に記載の数値制御装置において、外部より入力した文字コードをメモリに一担格納しておき、データ変換指令が出されたとき、実際のデータと変換することを特徴とする数値制御装置。

【請求項4】 数値制御装置内のデータを修正する際、実際のデータを修正することなく、修正情報を生成する修正情報生成手段と、この修正情報生成手段にて生成された修正情報を格納する修正情報格納メモリと、データ変換指令が出されたとき、上記修正情報格納メモリに格納されているデータを基に実際のデータを修正する手段とを備えてなる数値制御装置。

【請求項5】 数値制御装置内のデータを修正する際、実際のデータの修正が禁止されているか否かを判断し、その修正が禁止されていないとき、実際のデータを修正するとともに、その修正が禁止されているとき、実際のデータを修正することなく、修正情報を生成する修正情報生成手段と、この修正情報生成手段にて生成された修正情報を格納する修正情報格納メモリと、データ変換指令が出されたとき、上記修正情報格納メモリに格納されているデータを基に実際のデータを修正する手段とを備えてなる数値制御装置。

【請求項6】 請求項3～5の何れかに記載の数値制御装置において、実際のデータの修正可否をオペレータに判断させる手段を設けたことを特徴とする数値制御装置。

【請求項7】 請求項3～6の何れかに記載の数値制御装置において、修正履歴を格納する修正履歴格納メモリ及びこの修正履歴を表示部に表示させる手段を設けたことを特徴とする数値制御装置。

【請求項8】 請求項7に記載の数値制御装置において、指定された修正履歴を表示部に表示させる手段を設けたことを特徴とする数値制御装置。

【請求項9】 請求項7または請求項8に記載の数値制御装置において、修正履歴中の修正情報を編集する手段を設けたことを特徴とする数値制御装置。

【請求項10】 請求項7～9の何れかに記載の数値制御装置において、修正履歴中の修正情報を外部入出力機器または修正情報格納メモリに出力する手段を設けたことを特徴とする数値制御装置。

【請求項11】 数値制御装置内の同一名称のファイル

を、数値制御装置内部でディレクトリ情報の拡張子を用いることにより区分して扱う手段を備えてなる数値制御装置。

【請求項12】 請求項11に記載の数値制御装置において、数値制御装置内のデータを修正する際、実際のデータの修正が禁止されているか否かを判断し、その修正が禁止されているとき、同一内容の別ファイルを生成してこの生成したファイルを修正させるようにし、修正禁止状態が解除後、修正されたファイルを本来のファイルとして取り扱う手段を設けたことを特徴とする数値制御装置。

【請求項13】 請求項11または請求項12に記載の数値制御装置において、オリジナルファイルを保存し、所望時にそのオリジナルファイルを復元する手段を設けたことを特徴とする数値制御装置。

【請求項14】 次のステップを有することを特徴とする、変数を使用した加工プログラムを作成する数値制御装置の制御方法。

ステップ1) 実際の数値で定義された変数の付かない加工プログラムを作成するステップ、

ステップ2) ステップ1にて作成された加工プログラム上の、変数を用いて定義する部分を指定させるとともに、その指定箇所に使用する変数を定義させるステップ、

ステップ3) ステップ1にて作成された加工プログラム上の、ステップ2にて指定された箇所の実数の数値を、ステップ2にて定義された変数で置き換えるステップ。

【請求項15】 請求項14に記載のものにおいて、置き換える変数は、数式を用いることができることを特徴とする数値制御装置の制御方法。

【請求項16】 請求項14または請求項15に記載のものにおいて、変数名を定義する際、各変数に対応してメッセージデータを付加し、変数に値を入力する際、該メッセージデータを入力のためのガイダンスとして表示させることを特徴とする数値制御装置の制御方法。

【請求項17】 請求項14～請求項16の何れかに記載のものにおいて、変数の値が入力されていない場合、各変数を用いて定義した部分のデフォルト値によって加工プログラムのグラフィック表示を行うことを特徴とする数値制御装置の制御方法。

【請求項18】 請求項14～請求項17の何れかに記載のものにおいて、変数を用いて定義された加工プログラムの出力を行う際、加工プログラム中に変数で定義されている部分にマークを付加し、該マークと対応させて定義された変数内容を出力することを特徴とする数値制御装置の制御方法。

【請求項19】 加工プログラムの一部を変数で定義可能な数値制御装置において、該加工プログラムの変数で定義されている部分を指定された数値で置きえて表示する手段を備えてなる数値制御装置。

【請求項20】 加工プログラムの一部を変数で定義可能な数値制御装置において、該加工プログラムの変数で定義されている部分を指定された数値で置きかえた、新たな加工プログラムを生成する手段を備えてなる数値制御装置。

【請求項21】 加工プログラムの一部を変数で定義可能な数値制御装置において、1つ以上の変数の値を定義した複数のグループの各々に名称を付け、変数値を指定する際に、該グループ名称を指定することで、該グループに定義されている変数値を変数に設定する手段を備えてなる数値制御装置。

【請求項22】 自動プログラムデータを、外部入出力装置との間で、画面に表示されたデータ順に文字コードで入出力させる文字コード変換部を備えてなる数値制御装置。

【請求項23】 自動プログラムデータを、外部入出力装置との間で、各データ位置を特定する情報とともに文字コードで入出力させる文字コード変換部を備えてなる数値制御装置。

【請求項24】 数値制御装置内の各データに対してメモデータを設定する手段を備えてなる数値制御装置。

【請求項25】 請求項24に記載の数値制御装置において、予めメモデータ中に数値制御装置内のデータ許容範囲上限値と下限値とを設定しておき、この許容範囲内に各データが入っているかどうかを確認する手段を備えてなる数値制御装置。

【請求項26】 請求項24または請求項25に記載の数値制御装置において、あらかじめメモデータ中に数値制御装置内のデータのデフォルト値を設定しておき、所望時このデフォルト値に変更する手段を備えてなる数値制御装置。

【請求項27】 請求項23～請求項26の何れかに記載の数値制御装置において、データ設定時に各データに対応したメモデータの内容を表示させる手段を備えてなる数値制御装置。

【請求項28】 請求項23～請求項27の何れかに記載の数値制御装置において、メモデータを、データ部の表示画面番号及びデータの画面上の表示位置を特定する情報とともに入出力する手段を備えてなる数値制御装置。

【請求項29】 加工プログラムの解析を行い、該解析結果に基づいて機械を制御する数値制御装置において、該1ブロック単位の解析結果を予め記憶装置に格納しておき、該記憶装置より該1ブロック単位の解析結果を入力しながら機械を制御することを特徴とする数値制御装置。

【請求項30】 加工プログラムの1ブロック単位の解析結果を文字コードで入出力する手段を備えてなる数値制御装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 この発明は数値制御装置及びその制御方法に係わり、特に数値制御装置のデータ入出力制御に関するものである。

【0002】

【従来の技術】 数値制御装置（以下NCという）は紙テープ等から指令された加工プログラムに基づいて数値制御処理を実行し、該処理結果により工作機械を駆動してワークに指令通りの加工を施すものである。

【0003】 図60は、NC装置の要部ブロック図であり、1はNC装置、2はNC装置1に接続された外部の入出力装置である。NC装置1は、プロセッサ（CPU）10と、制御プログラム記憶用のROM14、RAM15と、ディスプレイ装置（CRT）19及びそのコントローラ（GDC）18、表示用のデータを格納しておくビデオRAM（VRAM）17と、キーボード（KEY）21及びそのコントローラ（キーボード制御）20と、各種パラメータやオフセットデータ等を記憶する不揮発性メモリ（バッテリーバックアップ用RAM）16と、各軸の軸制御部11と、所定のシーケンス処理を行って外部（機械側強電盤、操作盤）とデータ入出力を行うPMC装置12と、I/Oユニット13と、外部の入出力装置2とデータの入出力を行う入出力制御装置22から構成され、10、11、12、14、15、16、17、18、20、22の各要素はバスライン4で接続されている。

【0004】 図61は、NC装置1内に格納されている各種データの構成図であり、不揮発性RAM16内に格納されているものである。工具データは工作機械（図示せず）に装着されている工具（図示せず）のデータであり、工具形状データ91は工具の形状を示すデータを、工具補正量データ92は工具のノーズR補正值、工具オフセットデータ93は工具の取り付け位置を示すオフセット値を設定するものである。切削条件データ94は切削条件を自動決定する際に使用する値を設定するものである。加工プログラムデータはEIAで記述された加工プログラムを格納するエリア95と自動プログラムで記述された加工プログラムを格納するエリア96から構成されている。段取りデータ97は各加工で使用する爪形状のデータやワークの端面位置を示すZオフセット量等のデータが格納されている。パラメータ98はNC装置1で使用する各種のパラメータが格納されている。このうち、EIAの加工プログラム95のみ、文字コード（ASCII）で格納されている。

【0005】 図62は、NC装置の操作ボードの一例であり、CRT19及びキーボード21から構成されている。図63から図66は、CRT19上に表示されるNCの各種データの例であり、図63は、工具の現在位置等の情報を示す"POSITION"画面、図64は工具のオフセットデータを示す"TOOL DATA"画

5

面、図65は工具のノーズR等のデータを示す”NOS E-R”画面、図66はNC装置に格納されている加工プログラムの情報を示す”PROGRAM FILE”画面である

【0006】図67はNC装置1内の不揮発性メモリ16内に格納されているデータがCRT19上に表示された一例を示す説明図である。図67において、カッコ内の数値はCRT19上に表示された場合の行と列を示すものとする。すなわち、

— (5, 3) —

は5行3列目に表示されるデータであることを示すものである。

【0007】一般的にNC装置1内で各データを不揮発性メモリ16内に格納する場合にはデータタイプに合わせて格納する。すなわち、倍精度実数型のデータ (TYPE-L) は28に示すようにこのタイプのデータだけまとめて格納し、整数型のデータ (TYPE-N) は29に示すようにやはりまとめて格納される。

【0008】このように、不揮発性メモリ16内の各データはCRT19に表示される順と完全に対応して格納されているわけではないのが普通であり、データを表示する際にはデータを並べ換えて表示させている。

【0009】NC装置のデータタイプには、通常次のものが存在する。

倍精度実数型 : 8バイトのデータ 実数15桁を扱える

実数型 : 4バイトのデータ 実数7桁を扱える

倍精度整数型 : 4バイトのデータ 整数8桁を扱える

整数型 : 2バイトのデータ 整数4桁を扱える

【0010】従来、NC装置とデータの入出力を行おうとした場合には、簡単にはRS232C等のI/Fを用いて行っていた。加工プログラム95のように、EIAで記述された加工プログラムの場合、文字コードでデータの入出力を行うのが一般的である。これは、EIAの加工プログラム95ではNC装置内にデータを格納する際、文字コードのまま格納するため、容易に文字コードで入出力することができるためである。

【0011】近年では、NC装置に自動プログラム機能を内蔵するNC装置が多くなっている。図68は、加工図面の一例であり、ここで示されたワークを加工するための自動プログラムが図69に示すものである。図69で示す自動プログラムの内部データは特殊なデータ構造を有し、文字コードで格納されているわけではない。このため、自動プログラムのデータ96を入出力させる場合には、文字コードではなく、NC装置内部に格納されているデータをそのまま入出力していた。

【0012】加工プログラム以外にも、工具補正量92

6

やNC装置1のパラメータ98等を文字コードで入出力できる機能を有するNC装置も存在する。例えば、工具補正量92に関してはEIAの加工プログラム95と同様に、

G01L11 P__X__Z__Y__R__Q__;

ここで、P__ : オフセット番号

X__ : X軸オフセット量

Z__ : Z軸オフセット量

Y__ : Y軸オフセット量

10 R__ : 刃先R補正量

Q__ : 仮想刃先位置

のような形式の文字コードで入出力可能である。G10の指令により、工具補正量92の設定/修正を可能にするものである。

【0013】また、パラメータ98に関しても、

G10L50;

N__R__;

N__P__R__;

.

.

.

N__R__;

G11;

ここで、G10L50 : パラメータ入力モード

G11 : パラメータ入力モードキャンセル

N__ : パラメータ番号

P__ : 軸番号 (軸形パラメータ時)

R__ : パラメータ値

のような形式の文字コードで入出力可能である。

30 【0014】さらに、NC装置1と外部の入出力装置2とでデータの入出力を行う方法として、DNC機能を用いて行う方法があるが、これは専用のプロトコルを用いてデータの入出力を行うことになり、NC装置1及び外部の入出力装置2の相方でS/Wを専用に開発する必要がある。

【0015】次に、加工プログラムの処理方法に関して説明する。図70は、従来のNC装置1の加工プログラムの処理概要を示すブロック図である。

40 【0016】図70において、2は外部の入出力装置であり、フロッピーディスク、ICカード、カセットテープ等の制御装置、あるいはコンピュータシステムとの通信装置等がこれにあたる。22はNC装置1内の入出力制御装置、23は加工プログラム解析部、24はデータバッファ、25は機械制御部、26はサーボ制御部、27はサーボモータである。

【0017】NC装置1に対して、外部の入出力装置2より加工プログラムが入力される。この加工プログラムはNC装置1のメモリ16に一旦格納される場合と、そのまま解析処理部23に渡される場合がある。メモリ16に格納しきれないような大きな加工プログラムを実行

する際には、直接外部の入出力装置2から入力しながら加工プログラムの解析処理を行う。

【0018】加工プログラム解析処理部23では、加工プログラムのエラーチェックを行うと共に、加工プログラムの解析処理を行い解析結果をデータバッファ24に格納する。データバッファ24は加工プログラムの解析結果を一時格納しておくバッファエリアである。

【0019】機械制御部25は、データバッファ24より順次解析結果を取り出し、この解析結果により機械を制御する。この機械制御部において、補間演算を行い、この補間結果が各軸のサーボ制御部26に送られ、サーボ制御部26によって各軸のサーボモータ27がサーボ制御される。

【0020】金型加工等において、オフラインのCAMシステム等で生成した加工プログラムをNC装置1で処理する際、送られて来る加工プログラムは複雑な工具軌跡を微小線分で補間したデータであるので、極めて移動距離の短い指令が多量に連続するデータとなる。このため、工具の送り速度を早くしようとするNC装置の処理間が間に合わない場合があった。

【0021】これは、通常の加工プログラムの処理では加工プログラム解析処理部23が常に先行してデータを解析しており、データバッファ24に解析結果が格納されている状態となっているが、機械制御部25が解析処理部23が処理するよりも速くバッファ内のデータを消費し続けるとバッファ内のデータが空になってしまうためである。微小線分で補間したようなデータがかなりの数連続し、送り速度も速いとこのような状態になり易い。

【0022】NC装置の処理が間に合わなくなった場合、工具の送りが一時停止してしまい、加工ワークに悪影響を与えたり、加工時間が延びる等の不具合が生じる。このため、図71に示すようなバイナリフォーマットのデータでデータを転送する方法が提案されている。これは、4msec/8msecの工具の移動距離をバイナリデータで表現し、指定軸数分の移動量データによって高速処理で行わせるものである。指令軸数Nはパラメータで指定する。

【0023】1ブロック分のデータは $(2 * N + 1)$ byteである。各軸の移動量は全て2 byteで指令する。負の移動量は2の補数で指令する。CHECK_BYTEは、それ以外の $(2 * N)$ byteを1 byte単位で加算して、そのトータル値のうち、8 bit以上のオーバーフローを捨てたものである。

【0024】また、外部より入力される加工プログラムのフォーマットを制限することで高速処理を実現する方法も提案されている。加工プログラム中に高速区間を指令できるようにし、この区間内では指定可能なものは、
X: X軸の移動量
Y: Y軸の移動量

Z: Z軸の移動量

F: 切削送り速度

だけとしたものであり、この区間のデータは全て直線補間(G01)で加工される。

【0025】指令フォーマットは、

```

      .
      .
G05P01;      .....高速加工の開始
X__Y__Z__;
      .
      .
G05P00;      .....高速加工の終了
      .
      .

```

のように指定する。

【0026】

【発明が解決しようとする課題】

【0027】従来のNC装置1においては、外部の入出力装置2とデータの入出力を行おうとした場合、EIAの加工プログラムと一部のデータ(工具補正量、パラメータ)のみ文字コードで入出力可能であった。しかも、工具補正量、パラメータに関してはEIAの加工プログラムの一部という形式でのみ入出力可能であった。このため、外部の入出力装置2において取扱いが容易な文字コードでNC装置1とデータの入出力を行おうとした場合に制約を受けるという第1の問題があった。

【0028】NC装置1においては、実際に加工中であってもデータの設定を行える機能(バックグラウンド機能)を有するものがある。しかし、実際には実行中の加工に影響を与える工具補正量やパラメータの値は修正できず、修正可能なデータに制約があった。これは、パラメータや工具情報等の加工中に使用されているデータは、加工途中でむやみに修正されると危険であるからである。

【0029】しかし、次の工程で内部のデータの修正が必要な場合、従来では加工の終了を待ってデータを修正する必要があり、データ修正中は加工が行えず、機械の無駄時間を生じてしまう。

【0030】また同様に、バックグラウンド機能を用いて加工プログラムを修正する際にも、加工中の加工プログラムは修正できないという問題があった。

【0031】この問題を解決するために、「実開平1-127002号公報」に開示の考案のようにメモリを2重にし、このメモリを切り換えて使用する方法が提案されているが、この方式ではメモリエリアが倍必要となる。近年のNC装置のデータは非常に大量のものであり、この方式ではメモリの使用効率が極めて悪く、大幅なコストアップにつながってしまうという第2の問題があった。

【0032】また、NC装置1の工具補正量やパラメータ等を修正した場合、どのデータをどのように修正したかが分からないため、確認のため一つずつデータをチェックしなければならないという第3の問題があった。

【0033】なお、パラメータに関しては「実開平1-172133号公報」に開示の考案のように、修正以前のデータを記憶しておき、修正以前のデータを最も新しいものから表示する方法が提案されているが、これは表示するだけであり、しかもこの提示された方法によれば修正以前のデータを記憶しているだけであるので、修正前後のデータを比較することはできない。

【0034】また、従来のものは、誤って消去してしまったり、誤って修正してしまった加工プログラムを復元できなかった。

【0035】この問題を解決するために、「特開平3-116248号公報」には、一度削除したり更新したファイルを復元させるための方法が提案されているが、この方法ではファイルを管理する管理システム自体を改造しなくてはならず、システムの大幅改造となるという第4の問題があった。

【0036】また、NCに内蔵の自動プログラミングシステムで加工プログラムを定義する際、ほぼ同形状の加工ワークを加工する場合、加工プログラムは一部異なるだけでほとんど同じであるにもかかわらず、その都度加工プログラムを修正したり、各々別個に加工プログラムを登録しておく必要があるという第5の問題があった。

【0037】また、形状データに関しては、形状データを変数を用いて定義するパラメトリック形状定義方法があるが、これは専用の言語を用いて定義させる場合が多く、この専用言語を修得しないとパラメトリック形状を定義できないという第6の問題があった。

【0038】また、自動プログラムのデータは外部の入出力装置2で処理し易いような文字コードではなく、特殊なバイナリコードで入出力されるため、外部の入出力装置2上でデータ内容を見ようとしても専用のS/Wを開発しない限り不可能であるという第7の問題があった。

【0039】さらに、外部のCAM等で自動プログラムのデータを生成し、これを外部の入出力装置2からNC装置1に入力しようとしても、特殊なバイナリコードで入力する必要がありこれも専用のS/Wを開発しない限り不可能であるという第8の問題があった。

【0040】また、NC装置1に各データを設定する際、従来のNC装置では図64、図65のように”#1~#10”のような番号や”<X>、<Z>、<C>、<R>、<r>、<P>”のような略号で表示されるのみであるので、説明書等が無いとデータの意味が分からず、またデータの設定範囲や標準的な設定値も分からないという第9の問題があった。

【0041】また、NC装置1内の各データに不正な値

を設定してしまったり、不揮発性メモリ16の内容が何らかの原因で破壊された場合、検知できないという第10の問題があった。

【0042】さらにまた、外部のCAMシステム等で生成された微小線分から構成される加工プログラムを高速で実行しようとした際、一番高速で処理できるバイナリデータで転送しようとした場合、データ生成のために特殊なCAMシステムを作成しなくてはならないという第11の問題があった。

10 【0043】また、外部のCAMシステム等から微小線分で構成される加工プログラムを文字コードで転送する場合、高速モードを用いて転送すると記述内容が制限されるし、文字データ(10進数で記述)→バイナリデータ(2進数)

の変換処理を必要とするため、それほど高速化できないという第12の問題があった。

【0044】本発明は、上記のような第1の問題を解決するためになされたもので、NC装置と外部の入出力装置との間で、NC装置内の全てのデータに関して、入出力可能とし、またその入出力を文字コードで行うことが可能なNC装置を得ることを目的とする。

【0045】また本発明は、上記のような第2の問題を解決するためになされたもので、使用メモリ容量を大幅に増加させることなく、加工中において自由にNC装置内のデータを修正可能とし、かつ加工には何ら影響を与えないNC装置を得ることを目的とする。

【0046】また本発明は、上記のような第3の問題を解決するためになされたもので、NC装置内のデータを修正した場合、修正前のデータと修正後のデータを照合することが可能なNC装置を得ることを目的とする。

【0047】また本発明は、上記のような第4の問題を解決するためになされたもので、ファイル管理システム自体の大幅改造を伴うことなく、同一名称のファイルをNC内部で区分して取り扱えるNC装置を得ることを目的とする。

【0048】また本発明は、上記のような第5、第6、第7、第8の問題を解決するためになされたもので、加工プログラムのデータを変数を用いて定義可能とし、さらには、文字コードで入出力可能にしたNC装置を得ることを目的とする。

【0049】また本発明は、上記のような第9、第10の問題を解決するためになされたもので、NC装置内の各データを設定する援助機能を有するNC装置を得ることを目的とする。

【0050】更にまた本発明は、上記のような第11、第12の問題を解決するためになされたもので、外部より入力される加工プログラムのフォーマットを制限することなく、微小線分の高速処理を可能とするNC装置を得ることを目的とする。

【0051】

【課題を解決するための手段】

【0052】本発明に係わるNC装置は、NC装置内の各データを各画面対応で2次元もしくは1次元の配列データとして格納し、これらのデータを文字コード変換処理部で変換することにより、外部の入出力装置との間で文字コードで入出力できるようにしたものである。

【0053】また、本発明に係わるNC装置は、バックグラウンド中、もしくは特別に指定された場合には、各データの修正内容を示すデータのみを作成し、NC装置内の実際のデータは修正しないようにし、実際のデータが修正可能となった時点で作成した修正内容をもとに実際のデータを修正するようにしたものである。

【0054】また、本発明に係わるNC装置は、修正データをもとにNC装置内の実際のデータを修正する際、その都度オペレータの確認を得てから修正するようにしたものである。

【0055】また、本発明に係わるNC装置は、ファイルを管理する際、ディレクトリ情報の拡張子を用いて、同一名称のファイルを区分できるようにしたものである。

【0056】また、本発明に係わるNC装置は、加工プログラムのデータを変数で定義可能とし、加工プログラムを作成する際にこの変数の値を指定させるようにしたものである。また、自動プログラムのデータを文字コードに変換する機能、さらに、文字コードを自動プログラムのデータに変換する機能を持たせたものである。

【0057】また、本発明に係わるNC装置は、各データに対しての情報を付加できるメモデータを設定する機能を持たせるものである。

【0058】また、本発明に係わるNC装置は、加工プログラムの解析結果と等価のフォーマットのデータを直接NC装置に入力することで、加工プログラムの解析に要する時間を一切不要としたものである。

【作用】

【0059】本発明においては、NC装置内の全データが画面対応で何行何列目のデータという形で指定でき、各データは文字コードで外部の入出力装置との間で入出力可能となる。

【0060】また、本発明においては、NC装置が加工中で、本来であればデータの修正が不可能な場合等に、実際のデータを修正することなく、データの修正内容を示す修正データのみを作成する。

【0061】また、本発明においては、修正したいデータをオペレータに再確認させながらデータ修正を行う。

【0062】また、本発明においては、バックグラウンド編集において実行中の加工プログラムが格納されているファイルを修正する際には一旦そのファイルをコピーし、拡張子によって両者を区分できるようにし、コピーしたファイルを修正するようにさせ、加工プログラムが

修正可能な状態になった時に修正したファイルを本来のファイルとする。また、誤って消去してしまったり修正してしまったファイルも復元可能となる。

【0063】また、本発明においては、加工プログラムの一部を変数で定義できるようになったことで、同形ワークの加工プログラム作成が容易となり、自動プログラムのデータを文字コードで外部の入出力装置との間で入出力可能となる。

【0064】また、本発明においては、NC装置の各データ設定時にデータの説明を表示でき、データの範囲チェックやデフォルト値の設定を行うことが可能となる。

【0065】更にまた、本発明においては、加工プログラムの解析処理を行わないため、微小線分で構成される加工プログラムの高速処理が可能となる。

【0066】

【実施例】

実施例1.

【0067】以下、本発明の第1の実施例に関する一実施例を説明する。図1は、本発明に関するブロック図であり、NC装置1と外部の入出力装置2との間のデータの入出力に関しての処理を説明するものである。

【0068】外部の入出力装置2とは、一般的にペーパーテーブリーダ(PTR)やペーパーテーブパンチャ(PTP)、カセット(CMT)、フロッピーディスクドライブ(FP)等のことを指し、一般に市販されているパーソナルコンピュータ等でもさしつかえない。また、その他NC装置に接続可能な入出力装置であれば何でもよい。通常は、RS232C等のI/Fを用いるが、特にこれに限定するものではない。

【0069】32は、NC装置1内の各データが格納されているエリアであり、通常は図60における不揮発性メモリ16に格納されているデータである。33は、加工プログラム等のファイルとして管理されているデータであり、32と同様に不揮発性メモリ16内に格納されている。

【0070】30は、NC装置1と外部の入出力装置2間でデータの入出力を制御する部分であり、このデータ入出力制御部30に31の文字コード変換処理部が接続され、NC装置1内部の文字コード以外のデータを文字コードに変換したり、外部の入出力装置2から入力された文字コードをNC装置1内部のデータに変換する機能を有する。

【0071】32の内部データは、図2に示すようにNC装置1の各画面対応の配列データとして格納されている。例えば、図64に示す"TOOL DATA"は、図3に示すように縦10行横3列の配列データとみなし、32の内部データには図4に示すようなフォーマットで格納されている。図4に示したように、各画面データはヘッダ情報部とデータ部から構成され、ヘッダ情報は画面番号、データ部の配列の行数、列数及びデータ

タイプとからなる。

【0072】なお、画面番号は、本実施例では画面の通し番号で示すが、画面の区分に応じた番号にすることも可能である。例えば、工具情報に関する画面であれば順に、

"T1", "T2", "T3", ...

を画面番号とし、パラメータに関する画面であれば順に、

"P1", "P2", "P3", ...

を画面番号とすることも可能である。

【0073】要するに、画面番号とはNC装置1上に表示される画面を特定できる情報をもたせればよいということである。

【0074】行数、列数は図3で示したように、画面に表示されるデータの行数と列数を示すものであり、行数と列数を乗じた値がデータ部のデータ数となる。

【0075】データタイプとは、データ部に格納されているデータのタイプを示すものであり、例えば、

L: 倍精度実数データ

S: 実数データ

D: 倍精度整数データ

N: 整数データ

のように区分される。

【0076】本実施例においては、一画面分のデータを全て一つの配列データとして扱う例を示すが、一画面に種類の異なるデータを同時に表示するような場合には、図7に示すように右半分と左半分のようにデータを分けて複数の配列データに分割して格納することも可能である。この場合は画面番号を左半分のデータを示す配列を"2.1"とし、右半分のデータを示す配列を"2.2"のように付けて区分してもよい。

【0077】図7において、34はCRT19上に表示された左半分の表示データを示し、35は同じく右半分の表示データを示す。36は左半分の表示データ34に対応する配列データであり、37は同じく右半分の表示データ35に対応する配列データである。

【0078】また、図4に示した例によれば、

(1, 1), (1, 2), (1, 3), (2, 1),
(2, 2) ...

のような順にデータ部にデータが格納されている例を示したが、この他

(1, 1), (2, 1), ..., (10, 1), (10, 2) ...

のような順にデータ部に格納してもさしつかえない。

【0079】また、図63に示した"POSITION"画面のように配列で表現しにくい場合には、一次元の配列として図5に示すようなフォーマットで格納してもよい。

【0080】各々のデータは、画面番号と行、列で示し、例えば図64における

#3, <X>

に相当するデータ

"55.123"

は次のように示される。

(7, 3, 1) = 55.123

【0081】ここで、"7"は画面番号を示し、"3"が行番号、"1"が列番号を示すものである。

【0082】図4の配列データは、図6に示すような文字コードに変換される。あるいは、図6に示す文字コードを変換して、図4に示すような配列データが得られる。

【0083】図6において、

*H(7, 10, 3), L;

は、ヘッダ情報を示すものであり、

*END;

は、データの終わりを示すものである。

【0084】上記のように、本発明によれば図6に示したような文字コードにより、NC内部データ32を外部入出力装置に対して入出力することが可能となる。

20 【0085】なお、上記実施例では、図4に示すような配列データの構造を示したが、データの構造は特にこれに限定されるものではない。

【0086】実施例2. 次に、本発明による第2の実施例に関して説明する。図8は、オペレータ39がNC装置1上の操作ボード3を操作してNC装置の内部データ32を編集する場合のデータフローである。

【0087】まず、オペレータ39は、操作ボード3上のCRT19に表示されるデータを見ながらキーボード21を操作してデータの編集を行う。この際、CRT19上に表示されるデータは、内部データ32よりデータが取り出され(ルート1000)、このデータを表示処理部41がCRT19に表示できるように処理した後、CRT19に表示させる(ルート1001)。

【0088】オペレータ39がキーボード21より入力したデータは、編集処理部40に渡される(ルート1002)。

【0089】この場合、図9のフローチャートに示すように自動運転中等でデータの編集が禁止されているかどうかチェックする(ステップ100)。禁止されていないければ、内部データをそのまま編集する(ステップ101)。図8において、1003のルートである。これにより内部データ32の内容が変更される。データの編集が禁止されている状態とは、NC装置1が自動運転中で、実際に加工を行っており、内部データ32が変更されては危険な場合、もしくは、内部データ32を修正する際、実際に内部データ32を変更する前に修正情報を作成し、後から修正内容をチェックしながら修正する際に用いるデータ編集禁止モード時の状態である。

50 【0090】禁止されている場合、修正情報を文字コードに変換する(ステップ102)。この文字コードと

は、例えば、図64において、

#5, <Z>

の部分に

"111. 222"

というデータを設定しようとした場合、

(7, 5, 2) = 0, 111. 222;

という文字コードが生成される。

【0091】これは、第7画面の5行2列目のデータ"0"を"111. 222"に修正することを示すものである。すなわち、"()"内のデータが画面の表示位置を表わし、"7"が画面番号、"5"が行数、"2"が列数を示すものであり、"="の次が修正前のデータと修正後のデータを表わし、"0"が修正前のデータ、"111. 222"が修正後のデータを示すものである。

【0092】次に、この生成された文字コードを修正データ格納エリア38に格納する。図8において、1004のルートである。

【0093】次に、編集を行ったデータ部を反転表示させ(ステップ104)、どの部分が修正されたデータか一目で分かるようにする。

【0094】画面にデータを表示する際には、図8で示すように、表示制御部41は内部データ32(ルート1000)及び修正データ38を取り込み(ルート1005)、これらを合成してCRT19に表示するデータを生成する。すなわち、CRT19上には修正結果が表示される。

【0095】図12において、"TOOL DATA"の

#5の<X>に123. 456

<Z>に456. 789

のデータを設定する場合を考えると、51で示した画面がデータ修正前の画面であり、オペレータが52のように、

#(5) X(123.456) Z(456.789)C()

と入力してデータ入力を指令する"INPUT"キーを押すと、53で示すように画面上では設定したデータに修正されて表示され、仮に修正されたことを示す反転表示となる。

【0096】一つの画面で複数のデータを修正した場合には、図13で示すように修正部分が全て反転表示される。

【0097】図10は、修正データのみを反転表示させる処理を示すフローチャートであり、現在表示されている画面に対応する修正データが修正データ38の中にあるかどうかチェックし(ステップ110)、なければ反転表示は一切行われず、あれば対応するデータ部を修正データ値に修正して反転表示させ(ステップ111)、その他に対応するデータがあるかどうかチェックし(ステップ112)、あれば次々とデータを修正して反転表示を行う(ステップ111)。対応するデータが無くな

れば終了する。

【0098】データの編集禁止が解除された時点で、図8に示すように修正データを取り出し(ルート1006)、変更処理部42がこの修正データをもとに内部データ32を修正する(ルート1007)。修正が完了したデータ38は自動的に消去される。

【0099】以上の例はオペレータ39がキーボード21よりデータを入力する場合を示したが、データの編集が禁止されている状態で外部入出力装置2からデータを入力する際には、同様に一旦入力されたデータを修正データ38として格納し、データの編集禁止が解除された時点で同様に処理すればよい。

【0100】また、外部入出力装置2からデータを入力する際に、たとえデータの編集が禁止されていなくとも一旦入力したデータを修正データ38として格納し、変換指令が出された後にデータを修正するようにしてもよい。このようにすれば、実施例3で示すようにデータの修正を1つずつ確認しながら行うことも可能となる。

【0101】また、上記実施例では、修正したデータ部分を反転表示するようにしたが、特にこれに限定するわけではなく、修正部分が判別可能な別の方法を用いてもかまわない。例えば、修正部分に修正したことを示すマークを表示するか、カラーCRTであれば修正部分の表示色を変更する方法を用いてもよい。

【0102】また、履歴格納モードがONの場合、編集処理部40はデータの修正が行われた際、常に修正情報を文字コードに変換し、これを履歴データ45に格納する(ルート1012)。履歴データ45は修正データ38と全く同じフォーマットの文字コードで格納されており、修正順に記憶されている。

【0103】履歴データ45は割り当てられた記憶容量の許す限りデータを格納し、許容量を越えた場合には古いデータから順に消去し、最新の修正情報から記憶容量の範囲内の古い修正情報までが格納されている。

【0104】履歴データ45の内容はCRT19に表示させることが可能であり(ルート1015)、格納された順すなわち最新のデータから順に表示させることも、特定のデータ、例えば特定の画面データのみ表示させることも可能である(ルート1001)。

【0105】図15は履歴データ45をCRT19に表示させた例であり、(A)は最新のデータから順に表示させた例、(B)は画面7のデータのみ表示させた例である。画面下のメニュー表示は履歴データ45に対してどのような操作を行うかの選択用であり、"ALL"は履歴データ45を最新のデータから順に表示させるものである。1画面に表示できないデータは画面を切り換えて表示させることが可能である。

【0106】(A)の画面表示がこの例であり、画面右上の表示が現在表示されている画面が何画面中の第何画面かを示すものである。(A)の例では、5画面中の第

1画面を表示していることを示す。”SORT”は特定のデータのみ表示させるものであり、(B)の画面がこれに相当する。(B)の画面では第7画面に関する修正データのみ表示させた例である。

【0107】”EDIT”は履歴データを修正するものであり、図16(A)に示すようにカーソル46上のデータを自由に編集できる。編集は値を変更することやデータ自体を削除することが可能であるので、誤って修正してしまったデータの値を正しい値に修正したり、修正不要のデータを修正した場合にその修正データを削除することができる。また、修正前後のデータを入れ換えることで、一度修正したデータを再度復元するためのデータを作成することも可能である。(ルート1013、ルート1014)

【0108】例えば、

(7, 3, 1) = 55. 123, 64. 872

というデータがあった場合、このデータを修正前のデータに再度戻すために、修正前と修正後とのデータを入れ換え、

(7, 3, 1) = 64. 872, 55. 123

というデータに変更し、これを次に説明する”COPY”機能を用いて修正データに転送し、変更させることにより、(7, 3, 1)のデータを復元することが可能となる。

【0109】”COPY”は履歴データ45の内容の一部、または全てを修正データ38に転送するものである(ルート1016)。修正データ38に転送されたデータは以前に説明したように通常のルートで作成された内部データ32と全く等価に扱われる。

【0110】転送するデータの指定は履歴データ45の内容を全てコピーすると指定するか、図16(B)で示すようにデータの一部を指示して行うことができる。データの範囲の指定はコピーするデータの先頭と終わりとをキーボード21上のカーソルキーを操作して行う。指定されたデータはオペレータに明らかになるように反転表示される。”I/O”は履歴データ45の一部または全部を外部の入出力装置と入出力させるものであり(ルート1017)、出力の際には”COPY”の場合と同様に出力するデータを指定できる。

【0111】実施例3. 次に、本発明の第3の実施例に関して説明する。図8において、第2発明においては修正データ38を取り出し(ルート1006)、このデータをもとに修正処理部42により内部データ32を修正する(ルート1007)方式を示した。

【0112】本第3発明においては、修正データ38をもとに内部データ32を修正するのは同じであるが、修正を行う際に1つずつオペレータ39に確認をとってから修正を行う。

【0113】修正方法を図11のフローチャートを用いて説明する。まず、修正データ38が存在するかどうか

チェックする(ステップ120)。ここで、Nは修正データの数を示し、nは修正データのカウンターを示すものとする。

【0114】カウンター値を1に初期設定する(ステップ121)。画面番号Sを0に初期設定する(ステップ122)。Sは、CRT19に表示する画面の番号を示すものとする。

【0115】n番目のデータの画面NO. とSの値とを比較する(ステップ123)。図8ではルート1008である。n番目の画面NO. 値が異なっていた場合は、n番目の画面NO. を取り出し、Sに設定する(ステップ124)。画面番号Sの画面表示を行う(ステップ125)。

【0116】n番目のデータ部分を反転表示させる(ステップ126)。反転表示されているデータが修正対象のデータであることを示す。この場合、反転表示されるデータは、修正前のデータが表示されている。

【0117】データ入力部にn番目のデータを表示する(ステップ127)。図14における61がこの状態である。図8では、ルート1009がこれに相当する。

【0118】次に、オペレータに対してこのデータを修正するかどうか判断させる(ステップ128)。修正したい場合は、図62に示す操作ボード3上のキーボード21の”INPUT”キーを押し、修正が不要の場合には、”C. B CAN”キーを押す。図7では、1010のルートである。

【0119】修正の要求が行われた場合には、n番目のデータ修正処理を行う(ステップ129)。修正処理は、第2発明の場合と同様に、図8における修正処理部42が行う。ルート1011及び1007。修正後のデータは画面に表示される(ルート1009)。

【0120】n番目のデータ部分の反転表示をやめ(ステップ130)、次のデータの処理に移る(ステップ131)。

【0121】図14において、61の状態では”INPUT”キーを押す。すなわち、データの修正要求を出す。62の画面表示が行われる。修正要求のあった#5のデータは修正され、次に修正要求のある#7のデータ部に反転表示が移り、データを修正を行うかどうかオペレータの判断を待つ。

【0122】”C. B CAN”キーを押した場合は、修正をキャンセルすることになり、図14の63の画面表示のように#5のデータは修正されずに、次の修正要求のあるデータの処理に移る。

【0123】処理するデータが無くなれば全体の処理を終了し、あれば再びステップ123から処理を繰り返す(ステップ132)。

【0124】このように、本第3発明においては、修正データ38を1つずつオペレータ39が確認しながら内部データ32の修正を行うことができる。

【0125】オペレータがNC装置1の内部データ32を修正する際、後でもう一度確認しながら修正を行いたい場合は、データ編集禁止モードにしておいてからデータの編集を行い、データ編集禁止モードを解除してデータ確認モード（本発明の第3の実施例に係る発明の方式）で一つずつデータを確認しながらデータ修正を行うことが可能である。

【0126】データ編集禁止モードのON/OFF、及びデータ確認モードのON/OFF、履歴格納モードのON/OFFは、操作ボード3上のキーボード21の操作により行う。

【0127】実施例4. 次に、本発明の第4の実施例に関して説明する。図17において、70はファイルデータ33を管理するディレクトリテーブルである。71はファイル名称、72は拡張子、73はファイルの作成もしくは更新日付と時刻、74はその他のファイル管理情報である。

【0128】一般に通常のファイル管理では、ファイル名称の同一のものは必ず一つしか存在しないように管理しているが、本発明においては、72の拡張子を用いて同一名称のファイルを複数管理できるようにする。72の拡張子が“DAT”の部分は通常のファイルであり、今まで一般のNC装置1で扱われているファイルと同様のものである。図66にファイルのディレクトリ情報を表示させた例があるが、通常に存在するファイルとして扱われるのはこの拡張子72が“DAT”のファイルである。

【0129】一般のファイル管理システムにおいては、ファイルの区分はファイル名称と拡張子で行っている。よって同一名称のファイルでも拡張子が異なれば別ファイルとして扱える。本発明はこれを利用することにより、ファイル管理システムは従来のままの仕様とし、ファイル管理システムを利用するアプリケーション側のソフトウェアで上記のようなファイル名称の同一のファイル管理を行わせるものである。

【0130】拡張子72は、

BAK : 待機状態のファイル

OLn : 更新された古いファイル (n=1~9)

DAT : 通常ファイル

として区分される。

【0131】図18は、ファイル更新時のファイル状態を説明する説明図であり、今、ファイル名称“W100”のファイルを修正する場合を考えると、“W100”のファイルが編集禁止状態であった場合、すなわち、“W100”の加工プログラムが実行中の場合などは“W100”を修正することは不可能であるので、オリジナルの“W100”はそのままにして、修正を加えたファイルをファイル名称は同じ“W100”で、72の拡張子を“BAK”としたファイルとして別に作成する。オリジナルの“W100 (DAT)”はそのまま存

在するので、“W100 (BAK)”に対して自由に修正を加えることが可能となる。

【0132】“W100 (DAT)”のファイルが編集可能な状態の場合は、修正後のファイルを“W100 (DAT)”とし、オリジナルファイルの方は“W100” (OL1)として保存しておく。

【0133】このように、ファイルに対して編集を行うと、オリジナルファイルが常に保存されることになる。

【0134】編集禁止状態で生成された“W100” (BAK)のファイルは、編集禁止状態が解除された時点で

“W100” (DAT) → “W100” (OL1)

“W100” (BAK) → “W100” (DAT)

と修正され、修正後のファイルが通常ファイルとなり、オリジナルファイルは旧ファイルとなる。

【0135】図19はファイル操作を示すフローチャートである。ファイルの修正もしくはファイルデータを外部の入出力装置2から読み込んだ場合の処理を示すものである。

【0136】まず、同一のファイル名称71が存在するかどうか判別する（ステップ140）。存在しなければ、そのファイルを“DAT”の拡張子72を付けて格納する（ステップ149）。同一名称のファイルが存在した場合、ファイル名称71が同じで、拡張子72が“BAK”のファイルが存在するかどうか判別する（ステップ141）。なお、これより全てファイル名称71は同じファイルを扱うものとして説明する。存在しない場合には、拡張子72“DAT”のファイルをコピーして拡張子72“BAK”のファイルとする（ステップ142）。

【0137】なお、外部の入出力装置2からファイルデータを読み込む場合や、ファイルの編集を行う際に一旦全てファイルデータをワークエリアに転送し、これを修正後に元のファイルに格納するようなシステムの場合はこのステップは不要である。

【0138】拡張子72が“BAK”のファイルが存在する場合は、以前作成された拡張子72が“BAK”のファイルがまだ使用されていなかった状態であるので、そのまま修正を続けるものとする。

【0139】拡張子72が“BAK”のファイルに関して修正を加える（ステップ143）。指定されたファイル名称71の拡張子“DAT”のファイルが編集禁止状態であるかどうか判別する（ステップ144）。編集禁止状態であれば処理を終了する。この場合、拡張子72が“BAK”のファイルがそのまま残ることになる。編集禁止状態でなければ、拡張子72が“OLn”のファイルが存在するかどうか判別する（ステップ14

5）。“OLn”とは、

"OL1", "OL2", "OL3", ,
OL9"

の拡張子72を有するファイルのことである。

【0140】存在する場合、各々拡張子のnの値を1だけ増加させる(ステップ146)。すなわち、

OL3 → OL4

OL2 → OL3

OL1 → OL2

のように変更することを意味する。

【0141】次に、拡張子72が"DAT"のファイル10
を拡張子72が"OL1"のファイルに変更する(ステップ147)。さらに、拡張子"BAK"のファイル
を拡張子72が"DAT"のファイルに変更する(ステップ148)。これにより修正したファイルが正規の
ファイルとして登録されたことになる。

【0142】このようにすると、ファイル数がどんどん増加し、ファイルエリアが不足してしまうので、
ファイルエリアが不足した場合は、図20で示したフローチャートのようにファイルの解放を行う。

【0143】"OLn"ファイル、すなわちバックアップのため保存していた旧ファイルが存在するかどうか
チェックする(ステップ135)。“OLn”ファイルがなければ解放可能ファイルが無いので、
ファイル解放処理を終了し、“OLn”ファイルが存在すれば同一名称の“OLn”
ファイルが2つ以上存在するかどうかチェックする(ステップ136)。2つ以上存在する
場合には、そのなかで一番古い作成日付時刻のファイルを解放する(ステップ137)。
存在しない場合には、“OLn”ファイルの中で一番作成日付時刻の古いファイルを
解放する(ステップ138)。

【0144】このようにして不要ファイルの解放を必要に応じて行うことにより、
効率よくファイルエリアを使用するようにする。すなわち、本発明によれば、
ファイルエリアが不足しない限りオリジナルファイルを保存するようにし、
オリジナルファイルの復元が必要となった場合に対応可能としている。

【0145】図21は、ファイルのディレクトリ表示において、
ファイル復元表示モードにした場合の表示であり、この表示モードの場合には
図66で示した例と異なり、旧ファイル("OLn"ファイル)も含め全ての
ファイルが表示される。拡張子72を含めたファイル名が表示されるので、
全ての格納されているファイルを確認できる。

【0146】ここで、復元させたいファイル名をキーインすると
図22で示したフローチャートのようにファイルの復元を行う。

【0147】まず、指定されたファイル名が存在するかどうか
チェックする(ステップ150)。この場合のファイル名は図21の
<PROGRAM>で表示された拡張子72を含めたファイル名である。
指定ファイルが存

在しなければ処理を終了し、存在すれば次に、指定された
ファイル名が通常ファイル(拡張子72が"DAT"のファイル)のもの
かどうかチェックする(ステップ151)。通常ファイルであれば、
復元する必要はないのでそのまま終了し、旧ファイルの指定
(拡張子が"OLn"のファイル)であれば、同一名称の通常
ファイルが存在するかどうかチェックする(ステップ152)。

【0148】これは、例えば、

"100.OL2"

と指定された場合、

"100.DAT"

という通常ファイルが存在するかどうかチェックするものである。

【0149】存在する場合には、通常ファイルの拡張子と
指定ファイルの拡張子を互いに入れ換える(ステップ153)。
存在しない場合には、指定ファイルの拡張子72を"DAT"
に変更する(ステップ154)。

【0150】これは、例えば、

"100.OL2"

と指定された場合で、"100.DAT"が存在する場合
は次のようになる。

"100(DAT)" → "100(OL2)"

"100(OL2)" → "100(DAT)"

ここで、()内は拡張子72の内容とする。

【0151】以上により、一度消去してしまったファイルを
復元させたり、修正したファイルを修正前の状態に戻すことが
可能である。なお、本実施例においては、使用する拡張子72を

"BAK", "DAT", "OLn (n=1~9)"

30 で説明したが、特にこれに限定されるものではなく、
任意の名称の拡張子72を使用して差し支えない。

【0152】実施例5. 次に、本発明の第5の実施例に
関して説明する。図23(A)は旋盤において加工するワークの
加工図面である。横軸方向がZ軸、立て軸方向がX軸である。
いま、Z軸方向の2つの部分の長さを可変にしたい場合、
すなわち、図23(B)の"LA", "LB"で示した部分の長さを
変数で定義する場合の手順を図24で示したフローチャートで
説明する。

【0153】まず始めに、通常の自動プログラムの加工
プログラムを作成する場合と同様に加工プログラムを作成する
(ステップ201)。図25は図23(A)の加工図面から作成した
加工プログラムが表示されている画面である。画面の下に
表示されている“登録”, “印字”, “グループ化”, “変数化”は、
メニュー表示である。

【0154】次に、変数を用いて定義したい部分までカー
ソル46を移動させ、所望の位置にカーソルがきたら
メニュー“変数化”を選択する(ステップ202)。メニューの
選択は、メニュー表示に対応して設けられているメニュー
キーを押すことで行う。

【0155】変数式を入力する(ステップ203)。変数式は、変数と実際の数値、演算子(+, -, *, / 等)から構成される数式である。図25の例では、カーソル46位置に"LA"と入力する。入力された変数式の中に未定義の変数があるかどうかチェックし(ステップ204)、なければステップ206へ飛ぶ。未定義の変数とは、ステップ205を行っていない変数であり、何度も使用される変数は始めの一回だけ未定義ということになる。

【0156】未定義の変数に対しては、メッセージを定義する(ステップ205)。メッセージの定義とは、変数の値を定義する時、画面上に表示されるメッセージを定義することである。変数を全て定義し終わったら終了する(ステップ206)。

【0157】図26はこのようにして変数を全て設定し終わった後の画面表示であり、変数で定義した部分は反転表示されている(d1~d4)。元の数値は各変数で定義された部分のデフォルト値となる。

【0158】図23(B)のようにZ軸方向の長さを"LA", "LB"で定義する場合、d1~d4は次のように設定すればよい。

```
d1: LA
d2: LA
d3: LA+10
d4: LA+LB+10
```

【0159】このようにして加工プログラムの一部を変数化したものを登録する場合、図25で示した"登録"メニューを選択する。そして登録すべき加工プログラムの箇所をカーソル46で指定する。カーソル46を登録したい加工プログラム部分の先頭にまず移動させて"INPUT"キーを押し、続いて終わりの部分まで移動させて"INPUT"キーを押すことにより登録する加工プログラムの範囲を指定する。図27において、64の部分がカーソル46で指定した部分であり、指定部分は*

プロセス番号 シーケンス番号 データ位置

d1	:	1	1	6
d2	:	1	2	4
d3	:	1	2	6
d4	:	1	3	6

となる。

【0164】t15は各々の箇所で定義された変数式である。t16は元の加工プログラムの指定部分64、すなわち、変数で置き換える前の通常の加工プログラムがそのまま格納されている。

【0165】変数で定義された加工プログラムを確認したい場合、図25の"印字"のメニューを選択すると登録した変数付の加工プログラムが図29のようにプリンターに印字される。加工プログラムにおいて、変数で定義された部分は(n)で示され、その下に使用変数の一覧と変数で定義した部分の変数式が(n)に対応して印

*反転表示されている。登録する際には名称を付加して登録し、この名称によって登録した変数付加工プログラムを呼び出す。

【0160】図28は、上記のようにして登録されたデータ構造を示す説明図である。t1は登録する際に付加した名称であり、"BAR_TP16"と定義したことを示している。t4は使用されている変数の数であり、2つの変数が使用されていることを示す。t5は変数で定義された箇所の数であり、4箇所変数で定義されている部分があることを示す。t10は変数名であり、"LA", "LB"2つの変数が使用されている。t11は各変数のメッセージデータであり、

LA: STEP_1L

LB: STEP_2L

のようにメッセージが付加されていることを示す。

【0161】t12, t13, t14は変数で定義されている箇所を示すものであり、t12はプロセス番号、t13はシーケンス番号、t14は1行分のデータの前から順に数えたデータ位置を示す。これらは加工プログラムで作成する際画面上に表示されているものであり、図27において、"PNo."がプロセス番号、"SEQ"がシーケンス番号である。

【0162】図27の例ではプロセス番号は1、シーケンス番号は1~3であり、データ位置は、

- 1 形
- 2 前コーナ
- 3 始点X
- 4 始点Z
- 5 終点X
- 6 終点Z

となる。

【0163】図28の例では変数で定義された部分は4箇所であり、これは図26のd1~d4の位置を示し、各々

40 字される。

【0166】図23(A), (B)においては、加工プログラムの形状定義部(シーケンスデータ)のみを抽出する例を示したが、図68に示したような加工図において図30のように"1", "2"を変数化する場合を考えると、図31のように複数のプロセスに渡って変数で定義することになる。

【0167】図32は図31のデータ構造である。図31の例では"TB7025"の名称t1で変数t10"1"を"THR_L"のメッセージデータt11で、変数t10"2"を"LONG"のメッセージデ

ータ t11 で定義している。

【0168】このように、図27のように加工プログラムの一部、それもシーケンスデータのみを登録することも、図31のように加工プログラムの全体を変数で定義して登録することも可能である。

【0169】また、図33のように“@D1”，“@D2”部分の値は加工するタイプで5種類に固定されており，“LL”の部分だけその都度可変にしたい場合，“@D1”，“@D2”をグループ化して登録しておくことが可能である。

【0170】図34は図33を変数で定義した加工プログラムであり、変数のうち“@”が先頭に付加されている変数がグループ変数である。グループ変数はグループごとに値が決定される変数であり、図35はこのグループ変数を定義する画面である。

【0171】グループ変数の定義は図34で示したように加工プログラムを変数化した後に図25の“グループ化”のメニューを選択して行う。図35において、t17がグループ変数名であり、使用するグループ変数が表示される。t18はグループ名であり、使用したいグループ名を入力する。

【0172】図35の例では、5つのグループを定義し、各々の名称と数値は

TP1 : @D1=60, @D2=5

TP2 : @D1=70, @D2=5

TP3 : @D1=80, @D2=5

TP4 : @D1=100, @D2=7

TP5 : @D1=120, @D2=8

であり、例えばグループ“TP3”を選択すると変数“@D1”が80，“@D2”が5と指定されたのと等価となる。

【0173】このように、いくつかのタイプに設定値が固定される場合、このグループ化機能を用いると簡単にデータを設定できるようになる。図36は図34のデータ構造を示したものであり、t2はグループ変数の数、t3はグループの数であり、図34の例ではグループ変数の数は2、グループ数は5となる。t6はグループ変数でありt7はグループ名、t8はグループ変数“@D1”の各グループでの値、t9はグループ変数“@D2”の各グループでの値である。他は図32で説明したものと同様である。

【0174】次に、変数を用いて定義された加工プログラムを呼び出す方法に関して説明する。図33の加工を定義する際の例で説明する。図37は加工プログラムを定義する際の画面表示である。キーボード21よりデータを入力して行けば通常の加工プログラムの定義が行え、図68で示す加工図面をプログラミングすれば図43で示すような加工プログラムとなる。

【0175】加工プログラム作成中に変数で定義された加工プログラムを呼び出したい場合には、図37の下

メニュー”呼び出し”を選択する。

【0176】図38は変数型の加工プログラムを定義する際の処理を示すフローチャートである。まず、プログラム名称を入力させる（ステップ211）。指定されたプログラムが存在するかどうか判別し（ステップ212）、存在しなければエラー表示を行い（ステップ213）、終了する。指定プログラムが存在した場合、指定プログラムに対応するタイトル表示を行う（ステップ214）。

【0177】図39はタイトル表示の例であり、プログラム名称を“TBS7025”と入力し、各データを設定した後の画面表示である。“PARAM”の表示は変数型加工プログラムを意味し、その下にプログラム名称が表示される。図34の例ではグループ定義がなされているので、グループ名称を定義する“グループ”の表示が行われ、使用変数“LL”が表示されている。

【0178】次に、グループ指定がなされているかどうかチェックし（ステップ215）、グループ指定があればグループ名称を選択させる（ステップ216）。図39は“TP3”を選択したことを意味する。変数の指定があるかどうかチェックし（ステップ217）、あれば変数値を入力させる（ステップ218）。変数を入力させる際には、図39で示すように変数名が画面上に表示され、各変数に対応するメッセージが画面上に表示される。

【0179】例えば、LLの値を入力する際には、図36のt11で定義されているメッセージが“THR_L”

と表示されるので、どのような変数なのかオペレータが理解するのに役立つ。

【0180】変数型の加工プログラムがどのようなものであったか確認したい場合、図37で示した“形状表示”のメニューで選択すると図40に示すように“TBS7025”で定義された変数型の加工プログラムのデフォルト値で形状のグラフィック表示を行う。これにより、所望の加工パターンかどうか容易に確認できる。

【0181】形状のグラフィック表示は変数値を定義しなくてもデフォルト値で表示可能なため、変数型の加工プログラムを呼び出す場合、オペレータに対する援助機能として画面上に登録されている複数の変数型加工プログラムの形状を同時に表示させ、この中からオペレータが変数型の加工プログラムを選択するような機能をもたせてもよい。

【0182】また、図37で示した“数字表示”のメニューを選択すると、図39で定義した変数型加工プログラムが変数部分を定義された実際の値に変換されて表示される。図41は図39で定義された値を変換して表示させた例であり、変数で定義された部分は反転表示されている。

【0183】これは変数部分を実際の数値に変換して表

示しただけであるが、図37の“数値化”のメニューを選択すると、実際の数値に変換したプログラムを生成する。すなわち、図41で表示された加工プログラムが表示されるだけでなく、実際の加工プログラムとして存在することになり、図41の加工プログラムをキーボード21から直接入力した場合と等価の結果を得られる。一度変換した加工プログラムは自由に編集することが可能であるので、変数で定義されている以外の部分を修正したい場合に有効である。

【0184】また、図37で示した“文字出力”のメニューを選択すると、定義した加工プログラムを文字コードに変換して出力を行う。図42はこうして出力された加工プログラムの出力例であり、図69の加工プログラムを出力したものである。

【0185】自動プログラムのデータを入出力する際には、図42で示したように自動プログラム1行ごとに各々のデータをカンマで区切った文字データとして外部の入出力装置2と入出力を行う。出力されるデータは画面上に表示されるデータと同様のものである。

【0186】“W1100;”は加工プログラムの番号を示し、最後の“%”はデータの終わりを示す。自動プログラム各1行分のデータの終わりは“;”コードが付加されここまでが1レコードのデータである。

【0187】図69で示したプロセス番号(P N o.)、シーケンス番号(SEQ)は各々 P1, P2, P3, ... S1, S2, S3, ... と示される。

【0188】自動プログラムのデータは一部データを設定しなくともよい部分があるが、この部分は図42で示すようにただカンマのみ付加する。また、文字コードで表わせないような記号を自動プロが用いている場合、例えば、面粗さを示す三角マークなどは予め定めた文字データで代用する(“Z3”)。

【0189】なお、データが数値でなく、文字の場合には、データの前後を“”で囲んで示すものとする。例えば、“S45C”を示す場合には、“S45C”と記述する。

【0190】上記の例は、自動プログラムの1行分の全データを出力しなくてはならないフォーマットの例を記述したが、図43に示すように、各データの位置を示すようにしてもよい。図43の例では、自動プログラムの1行分のデータのうち、何番目のデータであるかを“@”で示すようにしている。

【0191】例えば、2番目のデータが“80.0”であることを示す場合には、@2=80.0と示せばよい。

【0192】この場合、不要なデータは省略可能とな

る。例えば、自動プログラム1行のデータの内の、1、2、5、6、9番目のデータのみ必要な場合、S1,@1=LIN,@2=5.0,@5=50.0,@6=45.0,@9=Z3;のように記述される。

【0193】この方式であれば、修正したいデータのみの指定可能であるので、本発明で示した第2、第3発明において自動プログラムのデータも処理できることになる。

【0194】また、以上説明したように、自動プログラムのデータも図32で示したように、プロセス番号t12、シーケンス番号t13、データ位置t14で各々のデータ位置を特定可能であるので、実施例1で示した内部データ32の処理と同様に図42、図43で示したようなフォーマット(文字コード)で加工プログラムの入出力が可能である。

【0195】実施例6.次に、本発明の第6の実施例に関して説明する。図44は、本発明によるメモデータに関して説明図である。メモデータは、NC装置1内の各データに対応して、設定許容範囲、標準設定値、表示シンボル、表示メッセージを指定するものである。

【0196】メモデータの設定は図44に示す表がNC装置1上のCRT19の画面上に表示されるので、この表にキーボード21よりデータを入力することにより行う。データを入力する位置はカーソル46をキーボード21上のカーソルキーを操作することにより指定する。

【0197】図44において、82が画面番号、83が行番号、84が列番号を示し、82~84で対応するデータを指定する。85は対応するデータの許容設定範囲の下限值、86は同じく上限値、87は標準設定値を指定する。88は対応データをCRT19の画面上に表示させる際のシンボルを指定し、89は対応データを入力する際にCRT19の画面上に表示されるメッセージを指定する。

【0198】図44の例では、第3画面の5行2列の位置にあるデータのメモデータを指定するものであり、データの設定許容範囲は0~20000であり、標準設定値は300で、このデータはCRT19画面上で“CLX”というシンボルで表わされ、このデータを設定する際には、

“X軸方向のクリアランス値”

というメッセージが表示されることを意味する。

【0199】NC装置1のセットアップ時や、データの内容が破壊された場合、一旦全データを標準設定値で設定させることが可能である。図46はメモデータに設定されている全データを標準設定値で設定し直す場合のフローチャートである。

【0200】まず、メモデータが指定されているかどうか判別する(ステップ401)。指定されていれば変数Nを1に初期化する(ステップ402)。変数Nはメモデータの順番を示す変数である。N番目のメモデータを

抽出する(ステップ403)。対応するデータ位置に標準値87を設定する(ステップ404)。対応するデータ位置は82~84で示されるデータの位置である。Nの値を1だけ増加する(ステップ405)。Nの値がNMAXの値を越えたかどうかチェックする(ステップ406)。NMAXはメモデータの設定個数である。残りのメモデータがあれば再びステップ403より繰り返す。

【0201】以上のようにして、メモデータに設定されている標準値87をメモデータで指定されている全てのデータに対して設定することができる。

【0202】NC装置1内のデータをCRT19上の画面に表示する際、メモデータ内のシンボル88を用いて表示する処理を図47のフローチャートにより説明する。

【0203】まず、表示する画面に対応するメモデータが存在するかどうか判別する(ステップ411)。対応するデータがあれば、メモデータより表示位置(83, 84)とシンボル88を抽出する(ステップ412)。表示位置(83, 84)にシンボル88を表示する(ステップ413)。

【0204】これは83の行番号84の列番号で示される位置に88のシンボルを表示することであり、図44の例であれば、5行2列の位置に"CLX"というシンボルを表示することになる。表示は図45の75で示すように行われ、図45の例では(5, 2)の位置のデータは"1520"という値であることが分かる。シンボルはデータを表示する際、単なるシリアル番号ではデータの意味がよく分からないため、番号の代わりに分かりやすいシンボルで表示させるためのものである。次に、その他の表示すべきデータがあるかどうか判別し(ステップ414)、あればステップ412より繰り返す。

【0205】以上により、データをCRT19上の画面に表示する際、メモデータ中に定義されたシンボル88を用いて表示することができる。

【0206】図48はNC装置1内のデータを設定する際の処理を示すフローチャートである。まず、カーソル46の位置を抽出する(ステップ421)。図45において46がカーソルであり、このカーソルが何行何列目のデータの位置に来ているかを抽出する。図45においては、5行2列がカーソル位置となる。

【0207】カーソル位置に対応するメモデータが存在するかどうか判別する(ステップ422)。図45が第3画面とすると、第3画面5行2列のデータがメモデータに設定されているかどうかを判別することを意味する。対応するデータがなければ、そのままデータをキーインさせ(ステップ423)、キーインさせたデータをそのままメモリに格納する(ステップ433)。すなわち、作成するメモデータが存在しなければ全く従来のデータ設定方法と同様となる。対応データが存在する場

合、メモデータより対応データを抽出する(ステップ424)。抽出したデータのメッセージデータ89を用いてCRT19画面上にメッセージ表示する(ステップ425)。

【0208】図44に示すように、
"X軸方向クリアランス値"

というメッセージデータが格納されていた場合、図45の76に示すように該メッセージデータがCRT19画面上に表示される。これによりオペレータは何のデータを設定しようとしているのかを明確に知ることができる。

【0209】次に、抽出したデータの許容範囲(85, 86)を用いて、CRT19画面上に許容範囲の表示を行う(ステップ426)。図44に示すように設定許容範囲の下限値が0で上限値が20000であれば、図45の77に示すように許容範囲が表示される。

【0210】次に、抽出したデータの標準値87を用いて、CRT19画面上に標準値の表示を行う(ステップ427)。図45の78がその表示である。次に、オペレータに設定データを入力させる(ステップ428)。標準値指定かどうか判別し(ステップ429)、標準値指定であれば抽出したデータの標準値87を入力されたデータとみなす(ステップ430)。標準値指定とは、図45の78で示す値を直接キーボード21から入力する代わりに、あらかじめ定められたキー(例えば"@"キー)を押すことにより代用させるものである。

【0211】例えば、図45において、実際の数値を入力する代わりに"@"を入力した場合、標準値300をキーボード21から設定した場合と等価に扱うことを意味する。標準値指定でなければ、入力されたデータが許容範囲内の値であるかどうか判別し(ステップ431)、許容範囲外であればCRT19画面上にエラー表示し(ステップ432)再度データ入力を行わせる。入力されたデータはメモリ内に格納されずに終わる。許容範囲内のデータであれば、入力データをメモリに格納する(ステップ433)。

【0212】図49は、NC装置1を起動させる場合やメモリ内に格納されているデータをチェックする際に行うデータチェックの処理を示すフローチャートである。まず、メモデータが格納されているかどうかを判別する(ステップ441)。メモデータが無ければ処理を終了し、有れば変数Nに1を設定する(ステップ442)。N番目のメモデータを抽出し(ステップ443)、メモデータの指定位置(82~84)のデータがメモデータの許容範囲内(85, 86)であるかどうか判別する(ステップ444)。範囲内であれば次のデータのチェックに移る(ステップ456)。範囲外であればメモリ内の設定データが不正であるので、データの再設定を行う処理を開始する。

【0213】まず、対応するデータ画面を表示させる

(ステップ445)。対応するデータ画面とは、メモデータの画面番号82で示される画面である。次に、メモデータの行番号83、列番号84で示される位置にカーソル46を表示する(ステップ446)。次に、図48で示したのと同様にメッセージ表示を行い(ステップ447)、許容範囲を表示し(ステップ448)、標準値を表示し(ステップ449)、データを再入力させ(ステップ450)、標準値設定であれば(ステップ451)標準値を入力データとし(ステップ452)、そうでなければ設定データのチェックを行い(ステップ453)、許容範囲外であればエラー表示し(ステップ454)、再度データ入力を行わせる。

【0214】許容範囲内であれば入力データをメモリに格納する(ステップ455)。Nの値を1だけ増加させ(ステップ456)、Nの値がNMAXを越えていないかどうか判別し(ステップ457)、越えていなければ再度ステップ443より繰り返すことで次のメモデータをチェックする。

【0215】以上の処理により、メモリに格納されているデータが不正な値をとっていないかどうかをチェックし、不正であればそのデータを表示し、再設定することが可能となる。

【0216】メモデータの内容は外部の入出力装置2に入出力することが可能であり、この際、次に示すような文字コードに変換して入出力可能であるので、メモデータの内容を容易に外部のシステムで表示、編集することが可能である。

【0217】図44のデータを文字コードに変換した場合、

(3, 5, 2) = 0, 20000, 300, "CL
X",

"X軸方向クリアランス値";

のように変換される。カッコ内の3つの数値が各々前から順に、画面番号82、行番号83、列番号84を示す。

ヘッダー部 データ部

$$(2+2+2) + ((2+4) * 30) = 186 \text{ (byte)}$$

の固定長のデータとする。ヘッダー部は6byte分のデータで、そのブロックデータがどのようなデータであるかを示す情報が格納されている。データ部は1~30個のデータであり、図52(B)で示すように、各々コード部(2byte)、データ部(4byte)から構成されている。

【0222】図53(A)で示すように、データ部の最後には必ず"EOB"データを付加する。"EOB"データ以降のデータは無効である。"EOB"データは図53(B)で示すような固定パターンのデータである。4byteのデータ部は符号付きの整数値で全て表わすものとする。

【0223】コード部は、図54(A)に示すようにフラグ部と識別コードから構成される。フラグ部の構成は

*す。(=)の次には、順に許容下限値85、許容上限値86、標準値87、シンボル88、メッセージ89が変換されて文字コードとなる。85~87は数値データであるので、10進数で示される数字で変換され、88、89は文字データであるので、文字データの前後を(")で囲って変換する。各々のデータの間はカンマ(,)で区切られる。

【0218】図50はメモデータを文字コードで出力させた例であり、先頭の、

*MEMO, 176;

はメモデータを示すものであり、"176"は出力するメモデータの総数を示すものである。

*END;

はデータの終わりを示すものである。なお、本実施例においては、メモデータの内容を図44に示すものとしたが、特にこれに限定されるものではなく、必要に応じて他の情報を付加しても差し支えないし、不要な情報は削除してもよい。

【0219】また、対応するデータを画面番号82、行番号83、列番号84で示すようにしたが、これも別の方法で対応するデータを示すようにしてもよい。メモデータの出力フォーマット(図50)も特にこれに限定するものではない。

【0220】実施例7。次に、本発明の第7の実施例に関して説明する。図51は、本発明によるNC装置1の加工プログラム処理概要を示すブロック図である。図51において、47は外部の記憶装置であり、フロッピーディスクやハードディスク等である。また、48はオフラインのCAMシステム等である。通常処理の場合には、図70で示した従来例と同様の処理であるが、データバッファ24のデータを図52(A)で示すフォーマットで格納するようにしている。

【0221】すなわち、1ブロック分の解析結果は、

図54(B)のようであり、識別コードがASCIIコードか特殊コードかの区分及び、コード部のデータ内容の区分を行うものである。フラグ部の最上位ビットがONの場合が識別コードが特殊コードであることを示す。OFFの場合がASCIIコードであることを示す。

【0224】図55はASCIIコードの一覧であり、図56が特殊コードの一覧である。一例として、図57(A)に示すような解析結果となった場合、図57(B)のようなデータが作成される。

【0225】本発明の実施例においては、加工プログラムを解析し、その解析結果をもとにそのまま機械を制御するだけでなく、コンパイルモード機能を有する。コンパイルモード機能とは、加工プログラムの解析結果を一且記憶エリアに格納しておき、実際に機械を制御する際

にはこの記憶エリアに格納されているデータをそのまま使用する方法である。

【0226】図58はコンパイルの方法を示すフローチャートである。まず、加工プログラムを読み出す(ステップ301)。加工プログラムの読み出しはNC装置1内のメモリ16からでも入出力制御装置22から入力しながら行ってもよい。次に、加工プログラムの解析を行う(ステップ302)。この解析結果を記憶エリアに格納する(ステップ303)。記憶エリアはNC装置1の内部メモリ(15または16)が十分な容量があればこれに格納してもよいし、外部の記憶装置47に入出力制御装置22を通して出力して格納してもよい。格納が完了したら(ステップ304)、解析が完了したかチェックし(ステップ305)、完了していればプログラムの解析を全て終了したかチェックし(ステップ306)、終了していれば処理を終える。こうして、加工プログラムの全解析結果が記憶エリアに格納される。

【0227】次に、コンパイルされた解析結果を実行する際の処理を図59のフローチャートで説明する。まず、コンパイルされたデータがあるかどうか判別する(ステップ311)。コンパイルされたデータがある時は、図58において説明した、加工プログラムの解析結果を格納したデータのうち、バッファ24に未転送のデータがあるかどうかということである。未転送のデータがあれば、データバッファ24に空があるかどうか判定する(ステップ312)。データバッファ24が1ブロック分のデータしか格納できないならこのデータが使用済みかどうか判別し、複数ブロック格納可能ならば、使用済みのデータもしくは空エリアがあるかどうかで判別する。バッファに空があればコンパイルデータを入力する(ステップ313)。

【0228】コンパイルデータの入力、図58で示したように、加工プログラムの解析結果を格納した記憶エリアから1ブロック分のデータを順に読み出すものであり、NC装置1の内部メモリ(15または16)、あるいは外部の記憶装置47に格納されているデータを読み出す。また、CAMシステム48において、本発明による解析結果と同様のデータを生成可能なものであれば直接外部の入出力装置2を通して入出力制御装置22から読み込むようにしてもよい。CAMシステム48で生成したデータが使用可能な場合、このデータを一旦外部記憶装置47に格納しておいてから必要なときに読み込ませることも当然可能である。

【0229】コンパイルデータがASCIIコードかどうか判別する(ステップ314)。通常コンパイルデータは図57(B)で示したようなフォーマットのバイナリデータであるが、NC装置1の外部に出力する際には図57(C)のようにバイナリデータをヘキサコードで示したASCIIコードで出力することが可能である。図57(C)において、一番号最後の";"は1ブ

ロック分のデータの最後を示すものであり、その前の"48"はそれより前の全ての1バイト分づつのヘキサコードを加算したチェックサムの値をヘキサコードで示すものである。

【0230】図57(C)の例では、

$$0 \times 80 + 0 \times 04 + \dots + 0 \times 00 = 0 \times 648$$

となるため、下2桁の"48"をチェックサムとしたものである。図51で示した文字コード変換処理部31において、図57(B)で示すような解析結果を図57

(C)で示すような文字コードに変換して出力するものである。このようにして、コンパイルデータ、すなわち、加工プログラムの解析結果は図57(B)で示すようなバイナリデータでも、図57(C)で示すような文字コードでも入力可能である。入力したコンパイルデータがASCIIコードならば文字コード変換処理部31でバイナリコードに変換し(ステップ315)、データをデータバッファ24に格納する(ステップ316)。

【0231】以上のようにして格納されている加工プログラムの解析結果を次々と読み出してデータバッファ24に格納し、通常処理と同様に機械処理部25はこのデータを読み出して機械を制御する。この際、加工プログラム解析処理部23は一切使用しなくてもよくなるので、加工プログラム解析に要する時間を完全に無くすることができ、データバッファ24にデータが格納されていないような状態を生じることを防止できる。

【0232】外部からデータをASCIIコードで読み込む場合も、入力されるデータがヘキサコードで表現されているデータであるので、通常の10進数で表現されているデータと比較してはるかに高速に変換可能となる。

【0233】なお、本実施例においては、加工プログラムの解析結果のフォーマットを図52～57に示すようなフォーマットで示したが、特にこれに限定されるものではない。また、出力フォーマットに関しても図57

(C)に限るものではなく、必要に応じて変更することは差し支えない。

【0234】

【発明の効果】本発明によれば、NC装置1の内部データ32を各々配列の要素として指定可能となり、また、外部の入出力装置2との間で文字コードで入出力可能となるので、NC装置1に接続された外部の機器で容易にNC装置1の内部データ32を扱うことが可能となる。

【0235】例えば、NC装置1に接続された外部の計測機器等でオフラインで計測を行い、この結果をNC装置にフィードバックさせようとした場合にも、従来ではデータの入出力に専用のS/W開発を必要としたが、本発明によれば文字データで内部データ32の入出力が可能であるのでデータの入出力に関する専用のS/W開発は不要となる。

【0236】また本発明によれば、使用メモリ容量を大幅に増加させることなく、NC装置1の内部データ32を機械が加工中であっても見かけ上変更可能であり、データが修正可能な状態になるとすぐに実際のデータを修正できるので、安価なシステムであるにもかかわらず、機械を無駄に遊ばせておく時間が大幅に短縮され、フォアグラウンドの状態に左右されないバックグラウンド処理が行える。

【0237】また、NC装置1に接続された外部の計測機器等でオフラインで計測を行い、この結果をNC装置にフィードバックさせる場合、従来ではデータをNC装置1の状態によってはデータを入力できない状態があった。すなわち、NC装置1が加工を実行中で内部データ32の修正が禁止状態にある場合である。よって、NC装置1の状態をチェックし、データの修正が可能になるまでデータ入力を待たせる処理が必要となるため、外部機器の時間の無駄となり処理内容も複雑なものであったが、本発明によれば、NC装置1の状態に係わらずデータを入力できるようになるので、安価なシステムであるにもかかわらず、外部機器の無駄時間はなくなり、また処理内容も簡単になる。

【0238】また、本発明によれば、NC装置1の内部データ32を修正する際、オペレータ39が一つずつデータを確認しながら修正することが可能となるので、データの修正ミス等が容易に見えける。

【0239】また、NC装置1の内部データ32を修正する場合、直接内部データを修正せずに、第2の発明のように修正データ38のみを一旦作成しこれをオペレータがチェックしながら実際のデータを修正することが可能となり、データの修正ミスを大幅に削減できる。

【0240】また、本発明によればファイル名の拡張子72の名称を操作するだけで、同一名称のファイルを内部で区分して取り扱えるので、専用のファイル管理システムを開発することなく、通常のファイル管理システムをそのまま使用して、NC装置1のファイルデータ33をNC装置1が加工中に修正したり、オペレータの操作ミス等で消失したファイルデータを容易に復元できるようになる。

【0241】また、拡張子72の名称変更はファイル管理システムに対して簡単なオペレーションで行え、処理も拡張子72の名称を変更するだけであるので高速に行える。

【0242】また、本発明によれば、自動プログラムのデータを変数を用いて定義可能となるので、NCの加工プログラムを定義する際、ほぼ同形状の加工ワークを加工する場合、その都度加工プログラムを別個に作成したり、各々別個に加工プログラムを登録しておく必要がなくなる。

【0243】また、NC装置1と外部の入出力装置2との間で自動プログラムのデータを外部で扱い易い文字デ

ータの形式で入出力可能となるので、外部で自動プログラムのデータを管理するのが容易となる。

【0244】また、外部のCAD/CAMシステム等で作成したデータを自動プログラムのデータに変換してNC装置1に入力することも容易となる。

【0245】また、本発明によれば、NC装置1内の内部データ32に対して、各々別個に情報を付加することが可能となり、データを設定する際にオペレータに対して適切な情報が提供できる。

【0246】また、データの設定範囲チェックを行えるので、不正な値を設定することを防止できる。また、標準設定値を設定できるので、システムの立ち上げ時などにとりあえず全てのデータを妥当と思われる値に設定しておき、動かしてみたい場合などに効果を発揮する。さらに、設定されているデータに対して範囲チェックを行い、不正な値が設定されていた場合にはそのデータを表示し、修正することが可能であるので、内部データ32が破壊された場合に容易に検知し、修正することができ

【0247】また、本発明によれば、加工プログラムの1ブロックの解析結果をそのまま入力可能であるので、外部より入力される加工プログラムのフォーマットの制限を無くして、加工プログラムの解析時間を不要とし、高速処理が可能となる。

【0248】また、入力フォーマットは専用のバイナリフォーマット以外に文字コードでも可能であるので、容易に外部のCAMシステム48等で入出力データを扱うことができる。さらに、解析結果のフォーマットは通常のNCの加工プログラム(EIA)と類似しているもので、解析結果に相当するデータを外部のCAMシステム48等で生成することも可能である。CAMシステムのポストプロセッサを本発明によるデータフォーマットで出力できるように変更することで、図51で示した加工プログラム解析部23を不要とするデータが生成できる。これにより、CAMシステムが生成した加工データをそのままNC装置1に転送し、NC装置1はこのデータをそのまま図51で示した機械制御部25に送って実行させるオンラインでの高速加工が実現可能となる。

【図面の簡単な説明】

【図1】本発明のデータ入出力部分の要部ブロック図である。

【図2】本発明の数値制御装置の内部データ説明図である。

【図3】本発明の数値制御装置の内部データ説明図である。

【図4】本発明の数値制御装置の内部データ構成図である。

【図5】本発明の数値制御装置の内部データ構成図である。

【図6】本発明の入出力データ説明図である。

【図 7】本発明の数値制御装置の内部データ説明図である。

【図 8】本発明のデータ入出力に関するデータフロー図である。

【図 9】本発明のデータ修正時のフローチャートである。

【図 10】本発明の修正データの画面表示に関するフローチャートである。

【図 11】本発明の修正データの画面表示に関するフローチャートである。

【図 12】本発明のデータ修正に関する操作説明図である。

【図 13】本発明のデータ修正に関する画面表示説明図である。

【図 14】本発明のデータ修正に関する操作説明図である。

【図 15】本発明の履歴データの表示例を示す図である。

【図 16】本発明の履歴データの表示例を示す図である。

【図 17】本発明のファイルデータ管理説明図である。

【図 18】本発明のファイルデータ処理説明図である。

【図 19】本発明のファイル操作を示すフローチャートである。

【図 20】本発明のファイル解放に関するフローチャートである。

【図 21】本発明のファイル解放の画面表示説明図である。

【図 22】本発明のファイル復元に関するフローチャートである。

【図 23】本発明の実施例の説明のための加工図面である。

【図 24】本発明の加工プログラムの変数化を示すフローチャートである。

【図 25】本発明の実施例の説明のための加工プログラムを示す図である。

【図 26】本発明の変数化後の加工プログラムの画面表示を示す図である。

【図 27】本発明の登録部分の指定を示す画面表示を示す図である。

【図 28】本発明の登録データのデータ構造図である。

【図 29】本発明の登録データの印字例を示す図である。

【図 30】本発明の実施例の説明のための加工図面を示す図である。

【図 31】本発明の実施例の説明のための加工プログラムを示す図である。

【図 32】本発明の登録データのデータ構造図である。

【図 33】本発明の実施例の説明のための加工図面である。

【図 34】本発明の実施例の説明のための加工プログラムを示す図である。

【図 35】本発明のグループ変数の定義画面を示す図である。

【図 36】本発明の登録データのデータ構造図である。

【図 37】本発明の加工プログラム定義画面を示す図である。

【図 38】本発明の加工プログラム定義のフローチャートである。

10 【図 39】本発明のタイトル表示例を示す図である。

【図 40】本発明の形状表示例を示す図である。

【図 41】本発明の実施例の説明のための加工プログラムを示す図である。

【図 42】本発明の自動プログラムの入出力データ説明図である。

【図 43】本発明の自動プログラムの入出力データ説明図である。

【図 44】本発明のメモデータ設定画面を示す図である。

20 【図 45】本発明のメモデータ表示例を示す図である。

【図 46】本発明の標準設定値で設定する処理のフローチャートである。

【図 47】本発明のデータ表示処理のフローチャートである。

【図 48】本発明のデータ設定処理のフローチャートである。

【図 49】本発明のデータチェック処理のフローチャートである。

【図 50】本発明の文字コード出力例を示す図である。

30 【図 51】本発明による NC 装置の加工プログラム処理の概要を示す要部ブロック図である。

【図 52】本発明のデータバッファ内のデータフォーマットを示す図である。

【図 53】本発明のデータバッファ内のデータフォーマットを示す図である。

【図 54】本発明のデータバッファ内のデータフォーマットを示す図である。

【図 55】本発明の実施例の説明のための ASCII コード一覧を示す図である。

40 【図 56】本発明の実施例の説明のための特殊コード一覧を示す図である。

【図 57】本発明の解析結果の内部構造を示す図である。

【図 58】本発明のコンパイル処理のフローチャートである。

【図 59】本発明のコンパイルデータの実行方法のフローチャートである。

【図 60】数値制御装置要部ブロック図である。

【図 61】数値制御装置内のデータ構成図である。

50 【図 62】従来の操作ボード図である。

【図63】従来の画面表示説明図 (POSITION) である。

【図64】従来の画面表示説明図 (TOOL DATA) である。

【図65】従来の画面表示説明図 (NOSE-R) である。

【図66】従来の画面表示説明図 (PROGRAM FILE) である。

【図67】数値制御装置要内のデータをCRTに表示する際の説明図である。

【図68】従来例を説明するための加工図面である。

【図69】従来例を説明するための自動プログラムデータを示す図である。

【図70】従来のNC装置の加工プログラム処理の概要を示す要部ブロック図である。

【図71】従来のバイナリフォーマットの例を示す図である。

【符号の説明】

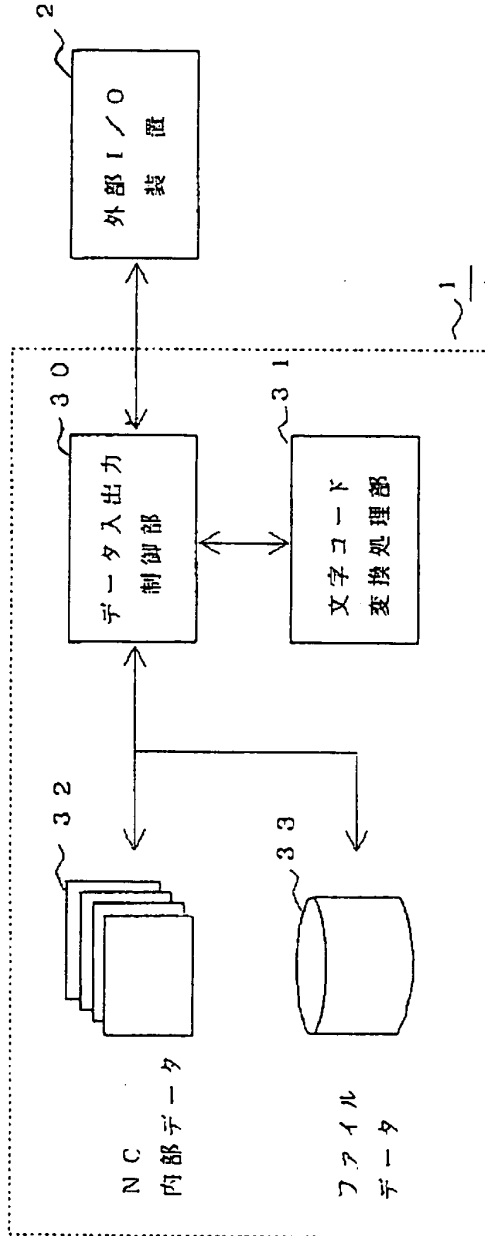
1. NC装置
2. 外部の入出力装置
3. 操作ボード
4. バスライン
10. プロセッサ (CPU)
11. 軸制御部
12. PMC
13. I/Oユニット
14. ROM
15. RAM
16. 不揮発性メモリ
17. VRAM
18. GDC
19. CRT
20. キーボード制御部
21. キーボード (KEY)
22. 入出力制御装置
23. 加工プログラム解析部
24. データバッファ
25. 機械制御部
26. サーボ制御部
27. サーボモータ
28. TYPE-Lのデータ
29. TYPE-Nのデータ
30. 入出力制御部
31. 文字コード変換処理部
32. 内部データ
33. ファイルデータ
34. 左半分のデータ表示
35. 右半分のデータ表示
36. 左半分の配列データ
37. 右半分の配列データ

38. 修正データ
39. オペレータ
40. 編集処理部
41. 表示処理部
42. 変更処理部
43. 確認処理部
44. 履歴処理部
45. 履歴データ
46. カーソル
- 10 47. 外部記憶装置
48. CAMシステム
51. 修正前の画面表示
52. 入力データ
53. 修正後の画面表示
61. 修正前の画面表示
62. 修正後の画面表示
63. 未修正後の画面表示
64. 指定部分
65. グラフィック表示
- 20 70. ディレクトリテーブル
71. ファイル名称
72. 拡張子
73. 日付時刻
74. 他の管理情報
75. シンボル表示
76. メッセージ表示
77. 許容範囲表示
78. 標準値表示
80. オリジナルファイル
- 30 81. 修正後のファイル
82. 画面番号
83. 行番号
84. 列番号
85. 許容下限値
86. 許容上限値
87. 標準値
88. シンボル
89. メッセージ
91. 工具形状データ
- 40 92. 工具補正量データ
93. 工具オフセットデータ
94. 切削条件データ
95. 加工プログラム (EIA)
96. 加工プログラム (自動プロ)
97. 段取りデータ
98. パラメータ
- t1 名称
- t2 グループ変数の数
- t3 グループ数
- 50 t4 変数の数

41

t 5 変数定義の箇所の数
t 6 グループ変数
t 7 グループ名
t 8 グループ変数値
t 9 グループ変数値
t 10 変数名
t 11 メッセージデータ

【図1】



42

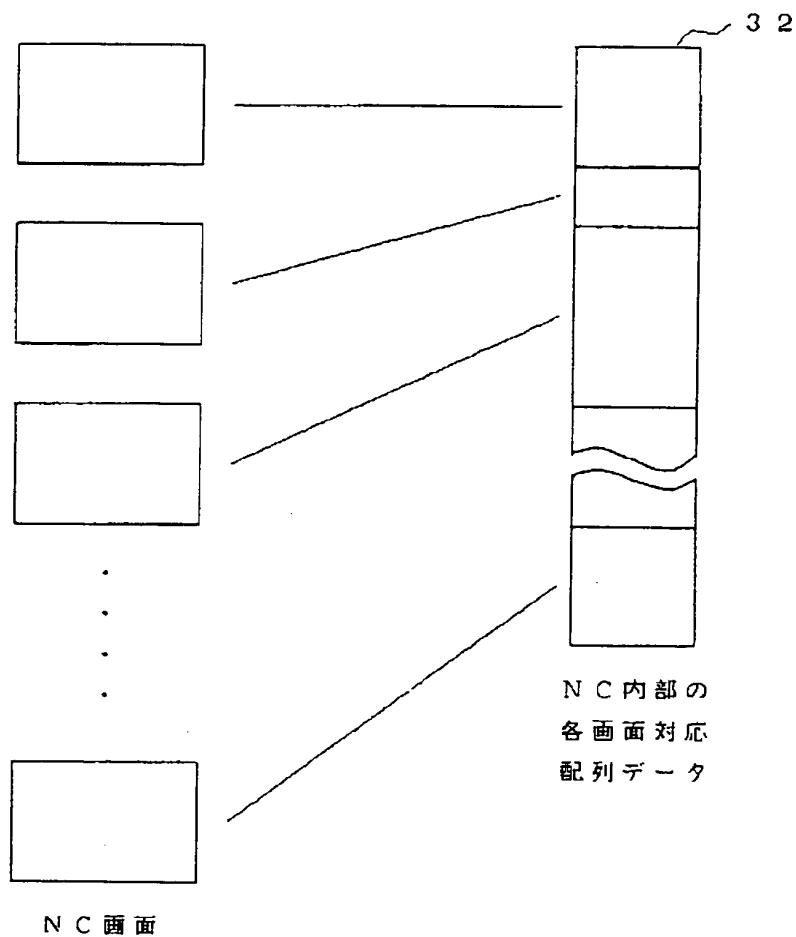
t 12 プロセス番号
t 13 シーケンス番号
t 14 データ位置
t 15 変数式
t 16 加工プログラム
t 17 グループ変数
t 18 グループ名

【図43】

```

W1100;
P0, @1="S45C", @2=80.0, @3=0.0, @4=155.0, @5=2000, @6=0.2, @7=0.1, @8=5.0;
P1, @1="EDG", @2="FCE", @3=110, @4=200, @5=0.3, @6=2.0, @7=1, @8=2;
S1, @1=80.0, @2=5.0, @3=0.0, @4=0.0, @5="Z3";
P2, @1="BAR", @2="OUT", @3=0, @4=80.0, @5=0.0, @6=130, @7=200, @8=0.3,
    @9=2.5, @10=3, @11=4;
S1, @1="LIN", @2=5.0, @5=50.0, @6=45.0, @9="Z3";
S2, @1="LIN", @2=5.0, @5=70.0, @6=70.0, @9="Z3";
S3, @1="凹", @3=70.0, @4=70.0, @5=80.0, @6=125.0, @9="Z3";
P3, @1="GRV", @2="OUT", @3=0, @4=1, @5=0, @6=10.0, @9=120, @10=0.08,
    @11=2.0, @13=5;
S1, @2=50.0, @3=45.0, @4=40.0, @5=45.0;
P4, @1="THR", @2="OUT", @3=0, @4=0, @5=2.0, @6=60.0, @7=1, @8=1.299, @9=10,
    @10=120, @11=0.3, @12=6;
S1, @1=50.0, @2=0.0, @3=50.0, @4=38.0;
P5, @1="END", @2=0, @3=0, @4=0, @5=0, @6=0, @7=0;
%
```

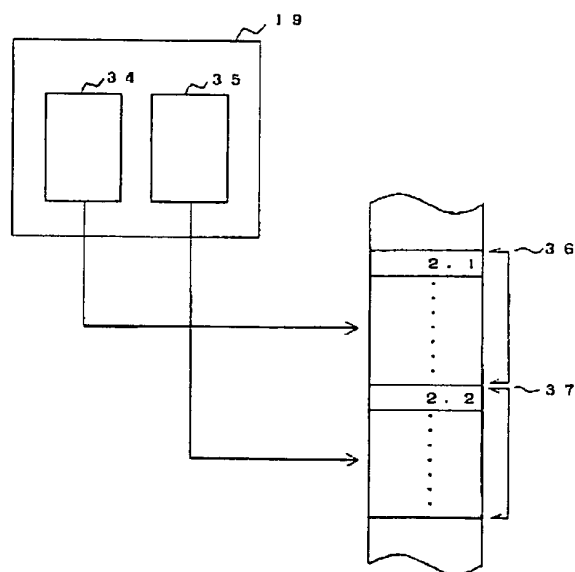

【図2】



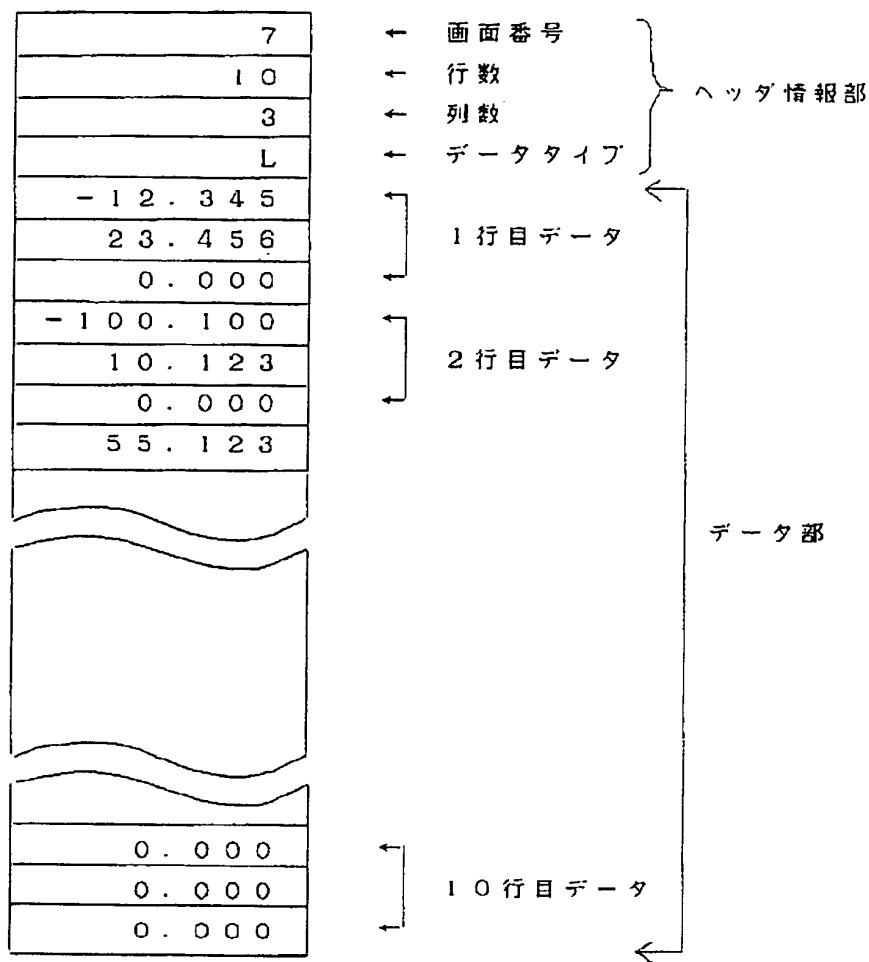
【図3】

	1	2	3
1	-12.345	23.456	0.000
2	-100.100	10.123	
3	55.123	100.234	
4	0.000	0.000	
5			
6			
7			
8			
9			
10	0.000	0.000	0.000
	↑ <X>	↑ <Z>	↑ <C>

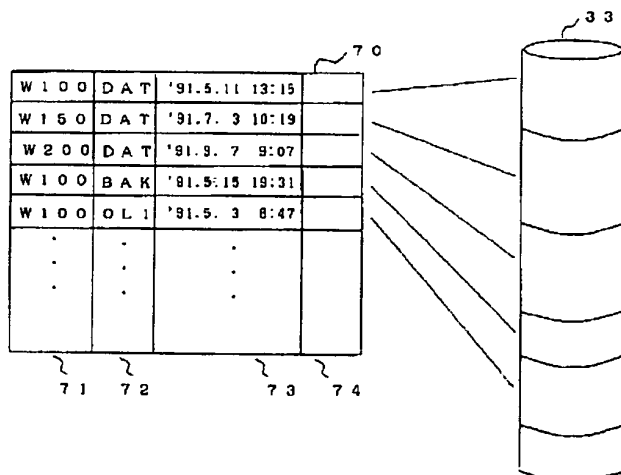
【図7】



【図4】



【図17】



【図21】

[PROGRAM FILE] IN/OUT 5. 1/1

PROGRAM ENTRY	8	REMAIN	32
CHARACTER	1500	REMAIN	16000
<PROGRAM> <CHR> <ST> <COMMENT>			
90.DAT	123	TEST CUT NO.12	
100.DAT			
200.DAT			
300.DAT			
400.DAT			
500.DAT			
100.OL1			
100.OL2			
RECOVER			
0(COMMENT()
INPUT	OUTPUT	ERASE	COPY
			FILE

【図5】

1	← 画面番号	} ヘッダ情報
1	← 行数	
1 4	← 列数	
L	← データタイプ	
1 2 3 4 5 6 7 8	← 0	} データ部
1 2 3	← N	
0	← -	
5 6 7 8	← 0	
1	← N	
1	← -	
1 2 3 . 4 5 6	← X	
3 4 5 . 6 7 8	← Z	
0 . 0 0 0	← C	
1 2 3	← S	
0	← T	
1 2 1 2	← M	
3 2 0	← S r p m	
0	← F c	

【図13】

[TOOL DATA]		PARAM 2.1/4	
#1: INC		[MACHINE]	
# A: ABS.			
#	(X)	(Z)	(C)
1	-12.345	23.456	0.000
2	-100.100	10.123	0.000
3	55.123	100.234	0.000
4	0.000	0.000	0.000
5	123.456	456.789	0.000
6	0.000	0.000	0.000
7	11.222	33.444	0.000
8	-55.666	-77.888	0.000
9	0.000	0.000	0.000
10	0.000	0.000	0.000
# () X () Z () C ()			
T-OFFSET	T-DATA	NOSE-R	PLC-SW MENU

反転表示

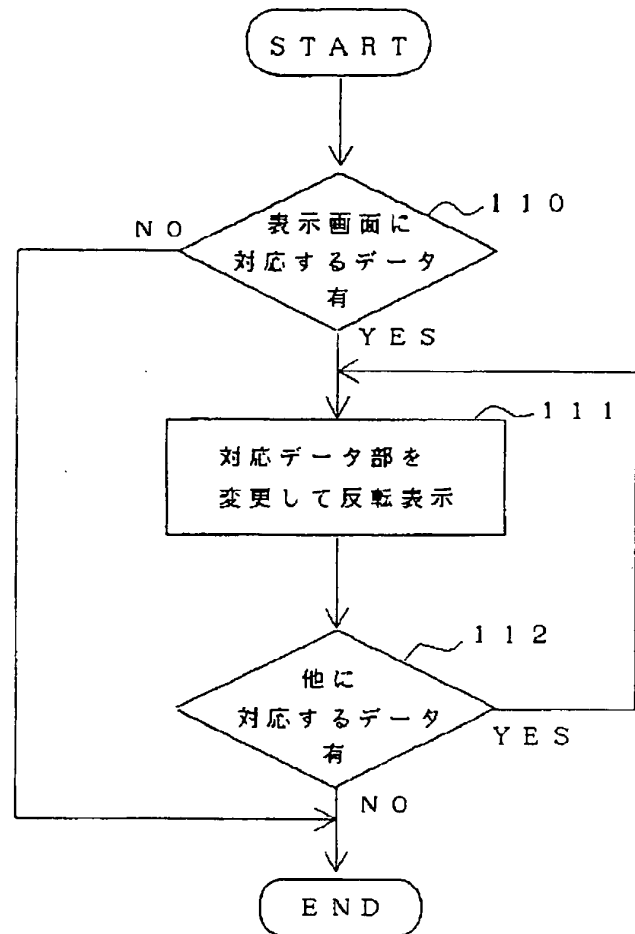
【図6】

```

*H (7, 10, 3), L;
(7, 1, 1) = -12.345;
(7, 1, 2) = 23.456;
(7, 1, 3) = 0.000;
(7, 2, 1) = -100.100;
(7, 2, 2) = 10.123;
(7, 2, 3) = 0.000;
(7, 3, 1) = 55.123;
(7, 3, 2) = 100.234;
(7, 3, 3) = 0.000;
.
.
.
.
(7, 10, 1) = 0.000;
(7, 10, 2) = 0.000;
(7, 10, 3) = 0.000;
*END;

```

【図10】

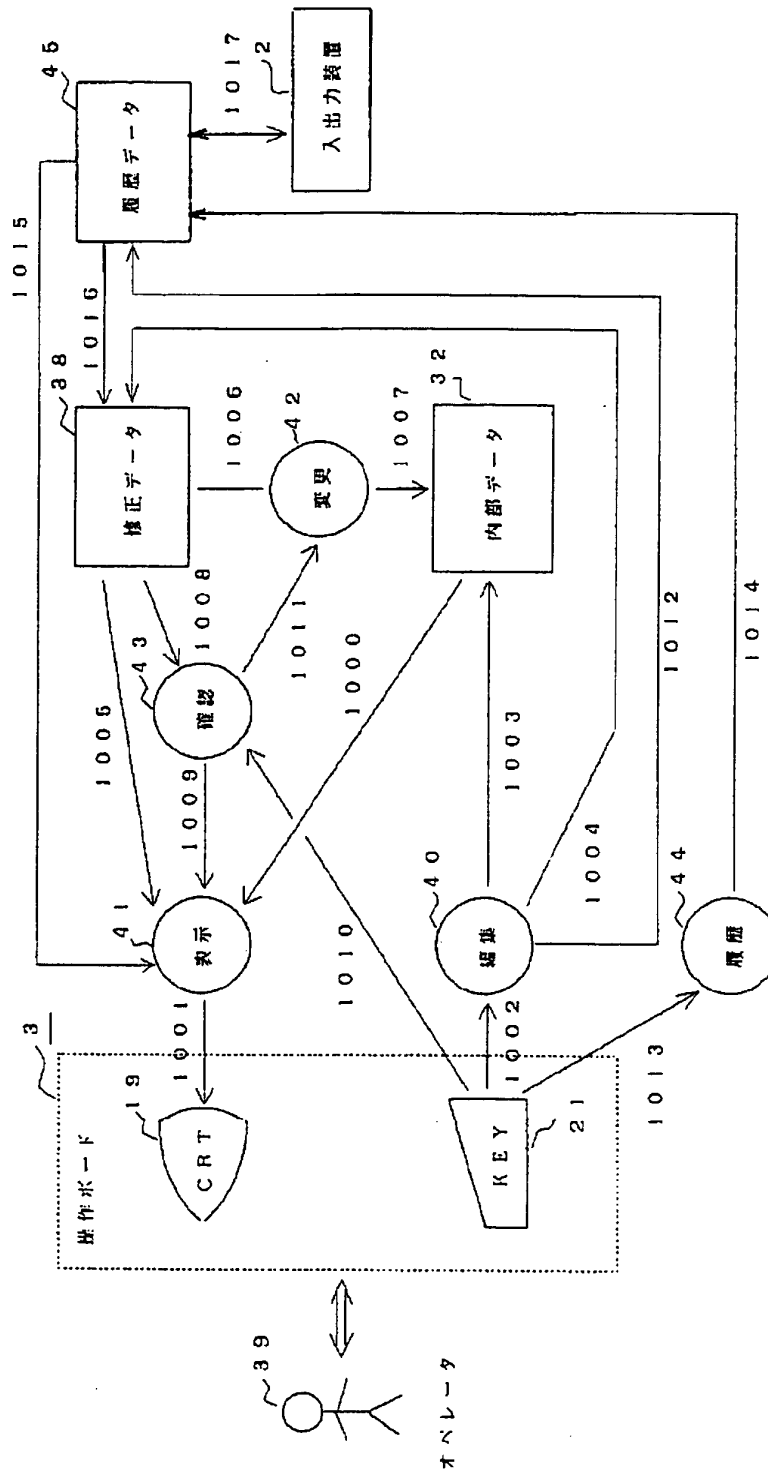


【図27】

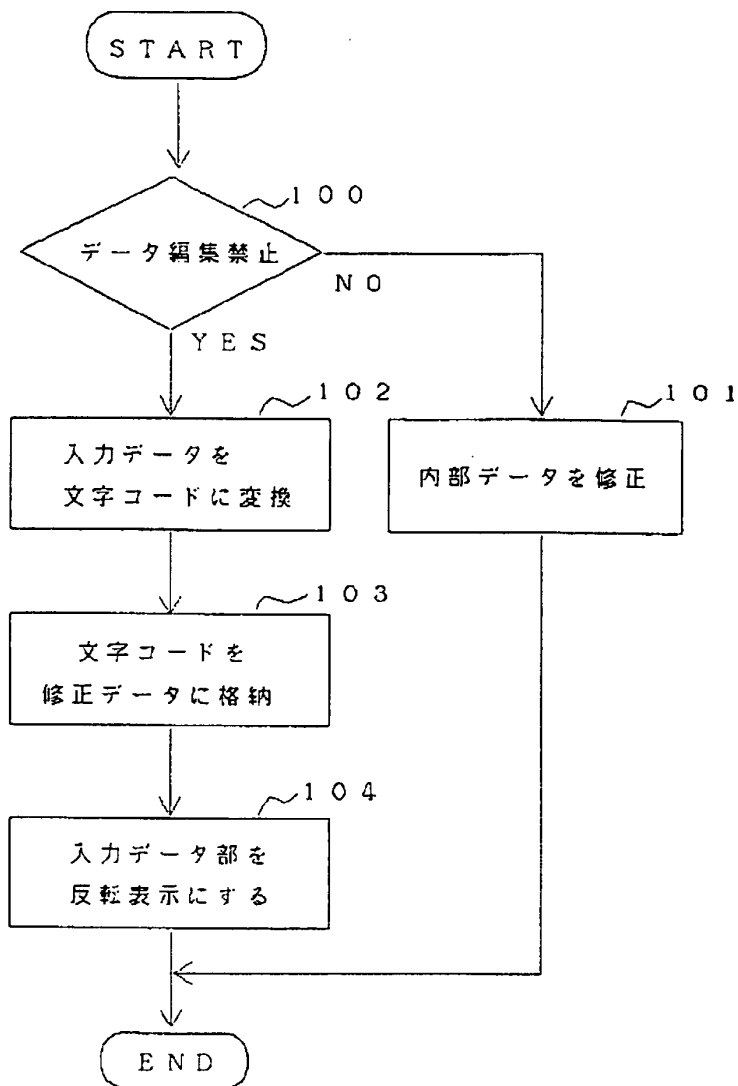
PNo.	材質	外径	内径	素材長さ	回転数	仕上X	仕上Z	素材端面			
0	S45C	120	0.	90.	2000	0.2	0.1	0.			
PNo.	モード	#	切込X	切込Z		R周速	F周速	R送り	R切込	R工具	F工具
1	BAR OUT	0	120.	0.		130	200	0.3	2.5	1	2
SEQ	形	前コーナ	始点X	始点Z	終点X	終点Z	後コーナ/\$	半径R/角度	粗さ		
1	LIN		◆	◆	60.	20.		◆	▼▼4		
2	TPR		60.	20.	100.	30.			▼▼4		
3	LIN		◆	◆	100.	50.		◆	▼▼4		
PNo.	モード	パーツカウント	復帰	ワークNo.	連続	回数	シフト量				
2	END	0	0		0	0	0.				

64

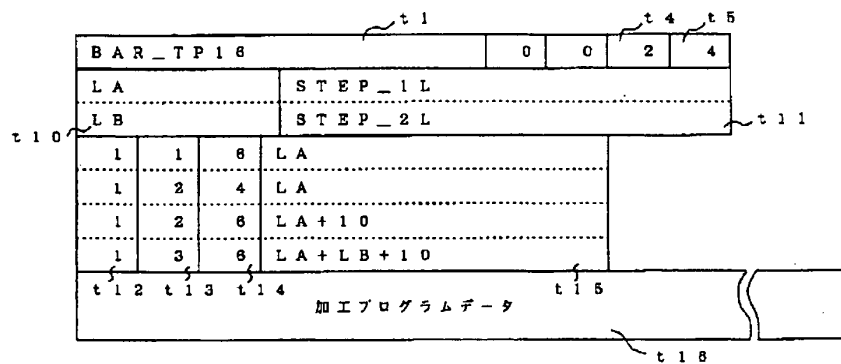
【図8】



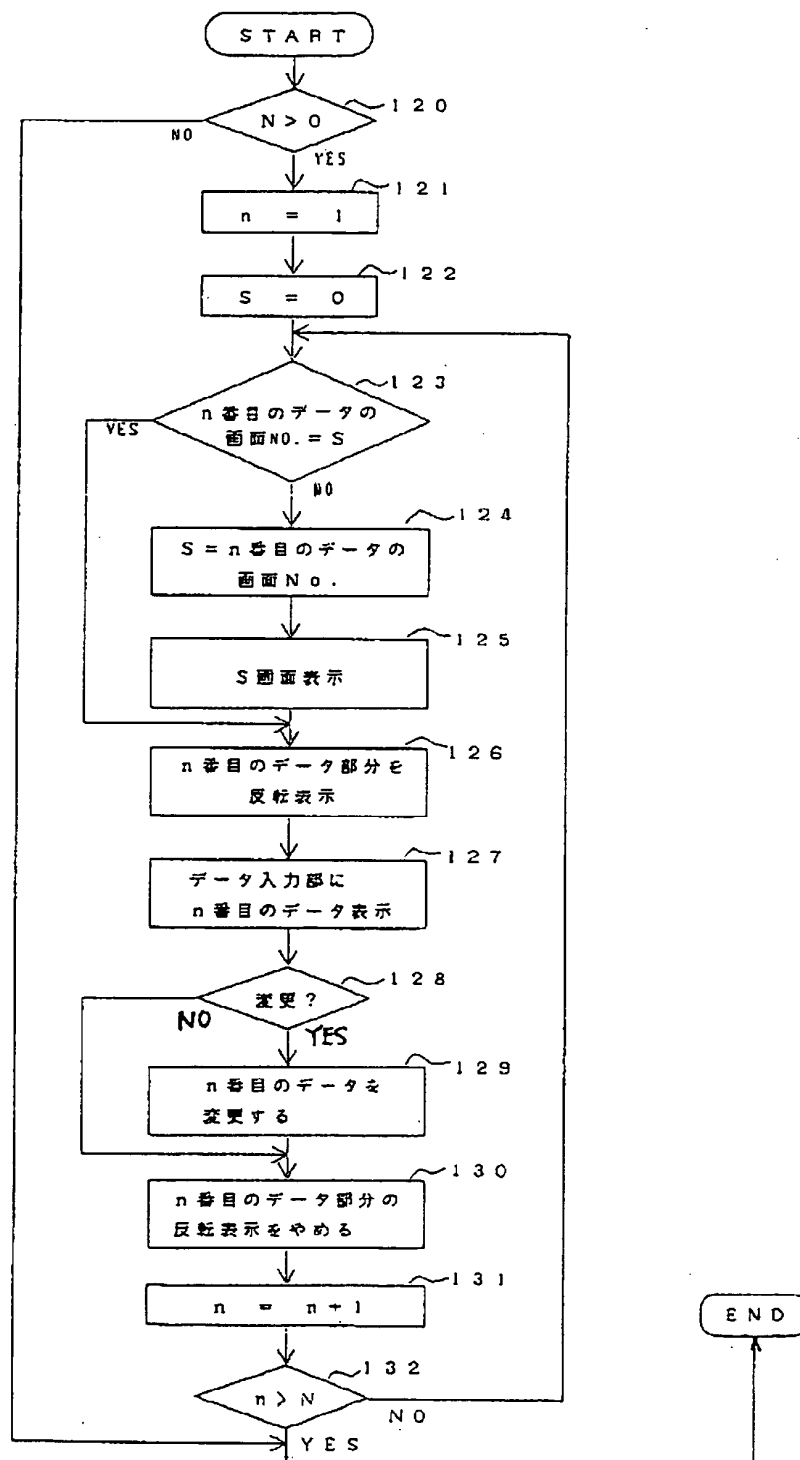
【図9】



【図28】



【図11】



【図12】

51

[TOOL DATA]		PARAM 2.1/4	
#1: INC.		#A: ABS. [MACHINE]	
#	(X)	(Z)	(C)
1	-12.345	23.456	0.000
2	-100.100	10.123	0.000
3	55.123	100.234	0.000
4	0.000	0.000	0.000
5	0.000	0.000	0.000
6	0.000	0.000	0.000
7	0.000	0.000	0.000
8	0.000	0.000	0.000
9	0.000	0.000	0.000
10	0.000	0.000	0.000

() X () Z () C ()

T-OFFSET **T-DATA** NOSE-R | PLC-SW | MENU

52

(5) X (1 2 3 . 4 5 6) Z (4 5 6 . 7 8 9) C ()

と入力して **INPUT CALC** キーを押す

53

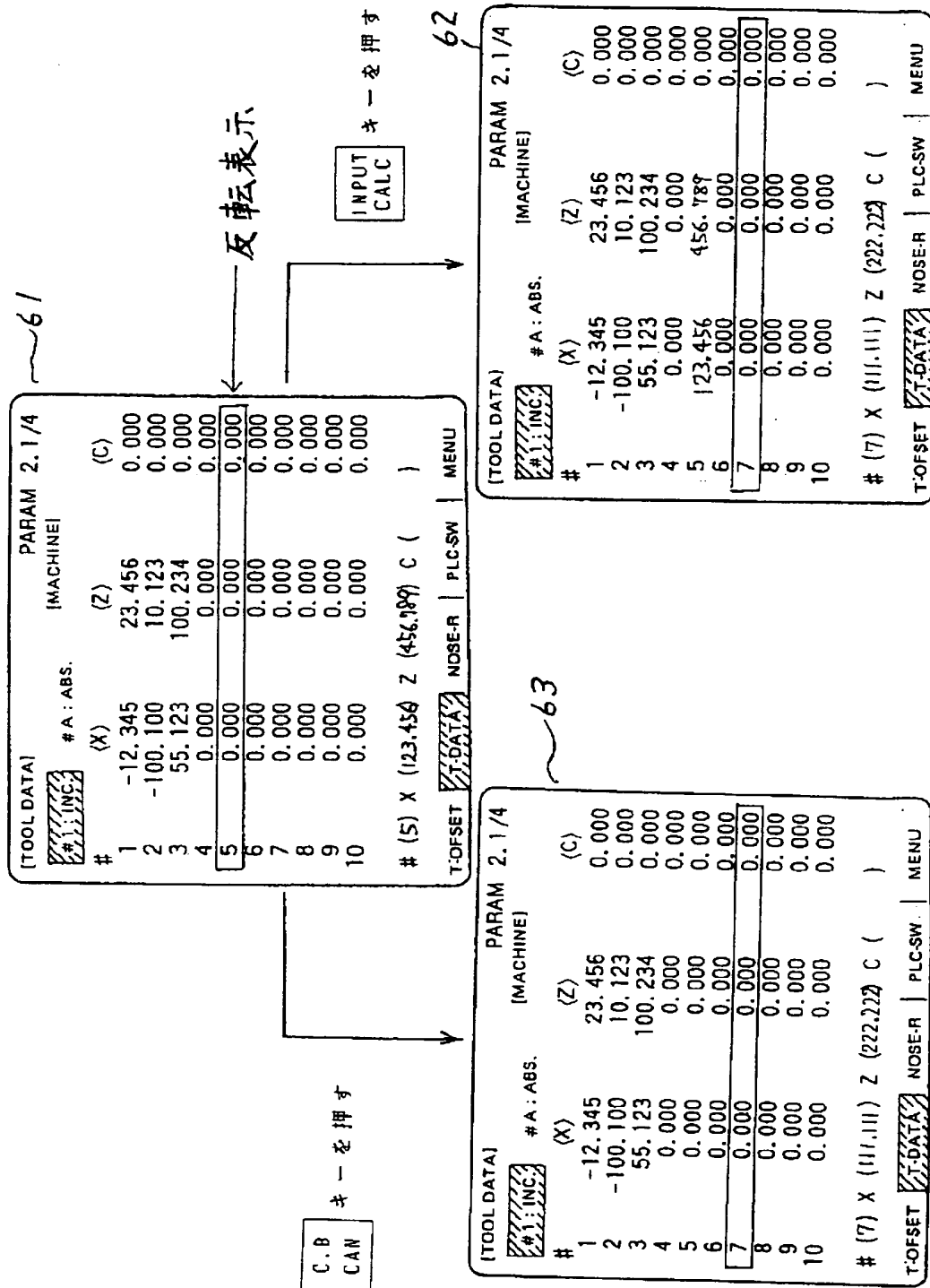
[TOOL DATA]		PARAM 2.1/4	
#1: INC.		#A: ABS. [MACHINE]	
#	(X)	(Z)	(C)
1	-12.345	23.456	0.000
2	-100.100	10.123	0.000
3	55.123	100.234	0.000
4	0.000	0.000	0.000
5	123.456	456.789	0.000
6	0.000	0.000	0.000
7	0.000	0.000	0.000
8	0.000	0.000	0.000
9	0.000	0.000	0.000
10	0.000	0.000	0.000

() X () Z () C ()

T-OFFSET **T-DATA** NOSE-R | PLC-SW | MENU

← 反転表示

【図14】



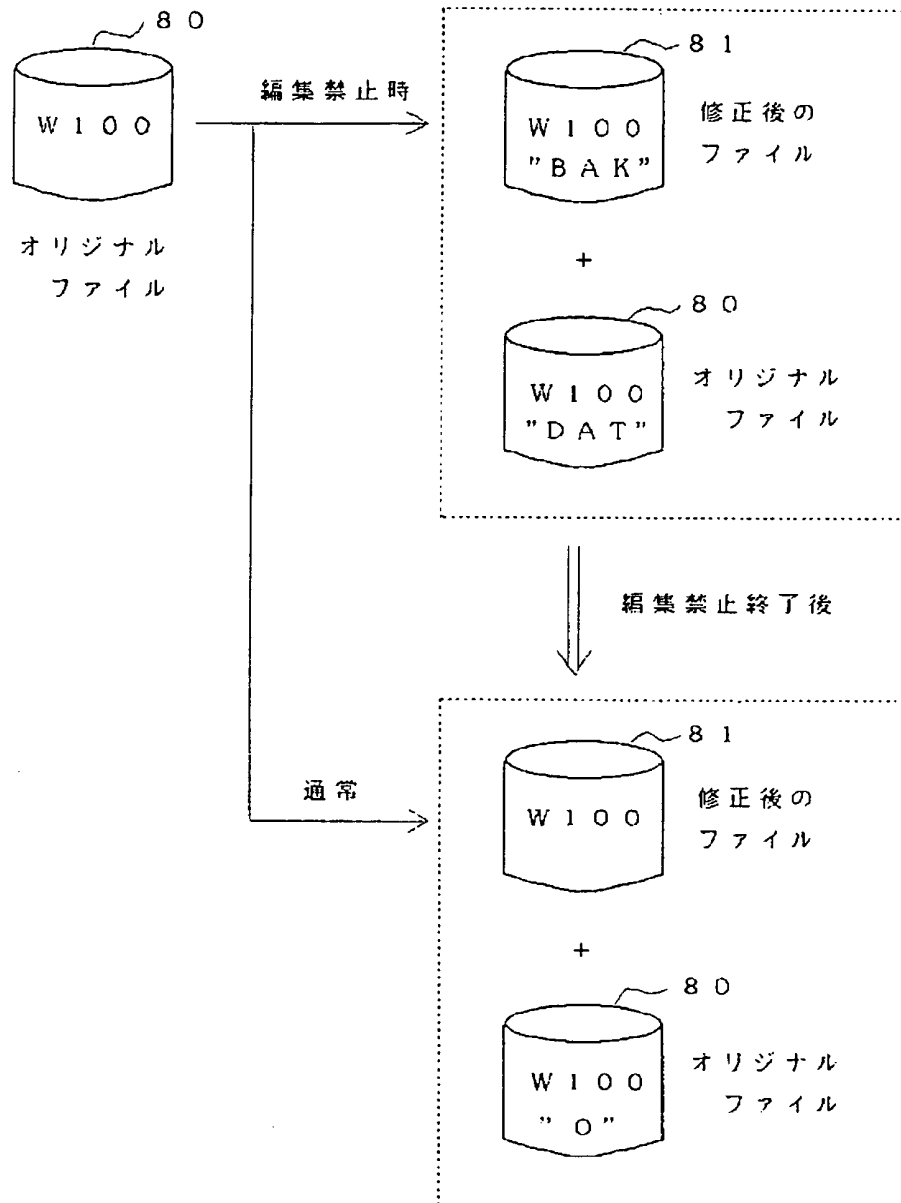
【図 15】

(A)		(B)	
19		19	
[MODI DATA]		[MODI DATA]	
1/5		1/2	
1. (7, 5, 2) = 0, 111, 222		1. (7, 5, 2) = 0, 111, 222	
2. (7, 3, 1) = 55, 123, 64, 872		2. (7, 3, 1) = 55, 123, 64, 872	
3. (5, 2, 3) = 19, 816, 21, 054		3. (7, 3, 3) = 15, 816, 21, 311	
4. (3, 1, 6) = 15, 3, 22, 7		4. (7, 2, 6) = 0, 213, 163	
5. .		5. .	
6. .		6. .	
7. .		7. .	
8. .		8. .	
9. .		9. .	
10. .		10. .	
ALL	SORT	EDIT	COPY
I/O			

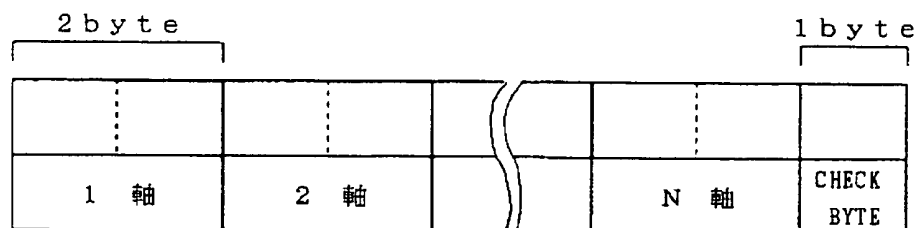
【図16】

(A)		(B)	
19		19	
[MODI DATA]		[MODI DATA]	
1/5		1/2	
1. (7, 5, 2) = 0, 111. 222		1. (7, 5, 2) = 0, 111. 222	
2. (7, 3, 1) = 55. 123, 64. 872		2. (7, 3, 1) = 55. 123, 64. 872	
3. (5, 2, 3) = 19. 816, 21. 054		3. (7, 3, 3) = 15. 816, 21. 311	
4. (3, 1, 6) = 15. 3, 22. 7		4. (7, 2, 6) = 0, 213. 163	
5.		5.	
6.		6.	
7.		7.	
8.		8.	
9.		9.	
10.		10.	
ALL	SORT	ALL	SORT
EDIT	COPY	EDIT	COPY
I/O		I/O	

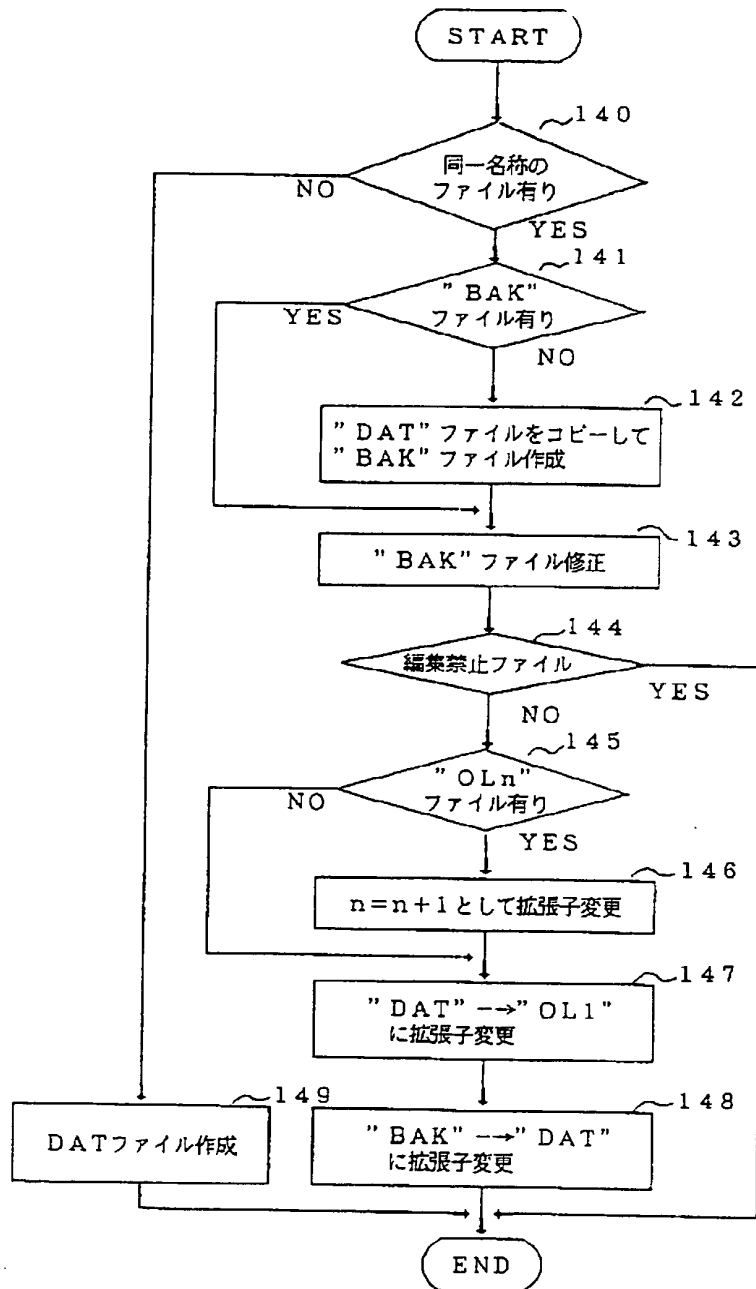
【図18】



【図71】



【図19】



【図63】

[POSITION]		MONITOR 1	
O12345678 N123-0			
<SUB>O 5678 N 1-1			
X	123.456	S	123
Z	345.678	T	M 1212
C	0.000 #1	Srpm	320
		Fc	0.00

POSI COORDI COMMAND SEARCH MENU

【図64】

[TOOL DATA]		PARAM 2.1/4	
#1: INC		#A: ABS. [MACHINE]	
#	(X)	(Z)	(C)
1	-12.345	23.456	0.000
2	-100.100	10.123	0.000
3	55.123	100.234	0.000
4	0.000	0.000	0.000
5	0.000	0.000	0.000
6	0.000	0.000	0.000
7	0.000	0.000	0.000
8	0.000	0.000	0.000
9	0.000	0.000	0.000
10	0.000	0.000	0.000

() X () Z () C ()

T-OFFSET T-DATA NOSE-R PLC-SW MENU

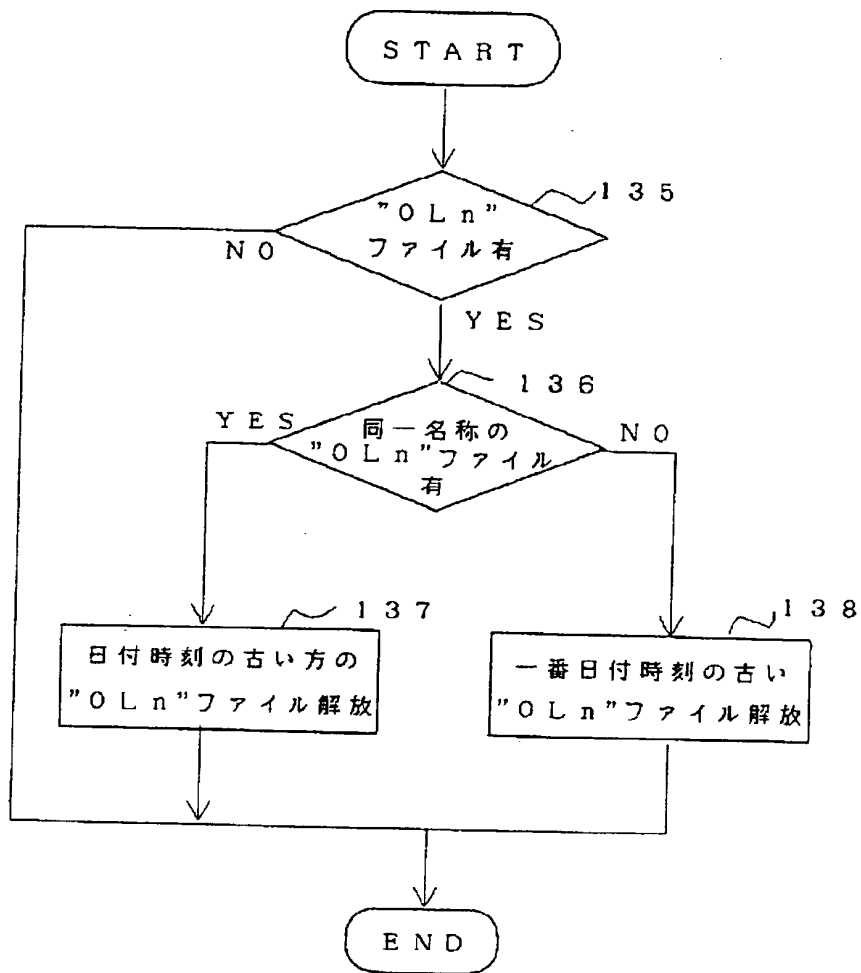
【図65】

[NOSE-R]		PARAM 3.1/4	
#	(R)	(r)	(P)
1	5.000	0.045	3
2	10.000	0.099	8
3	6.000	0.099	2
4	0.000	0.000	3
5	0.000	0.000	3
6	0.000	0.000	3
7	0.000	0.000	3
8	0.000	0.000	3
9	0.000	0.000	3
10	0.000	0.000	3

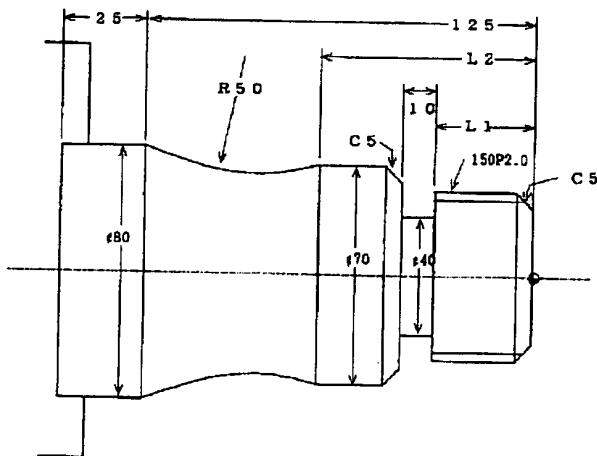
() R () r () P ()

T-OFFSET T-DATA NOSE-R PLC-SW MENU

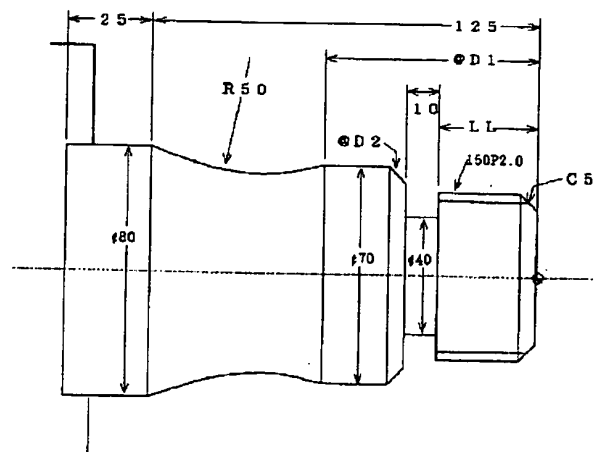
【図20】



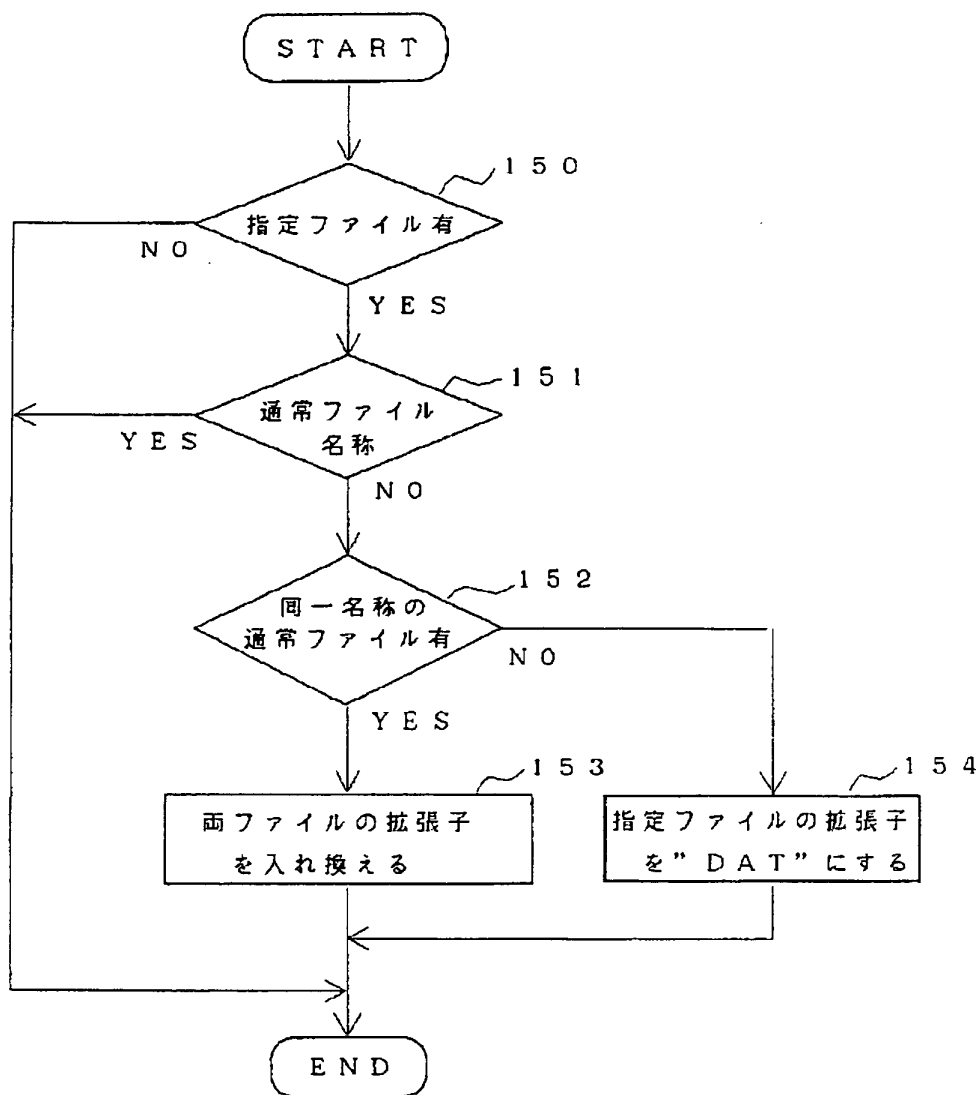
【図30】



【図33】



【図22】

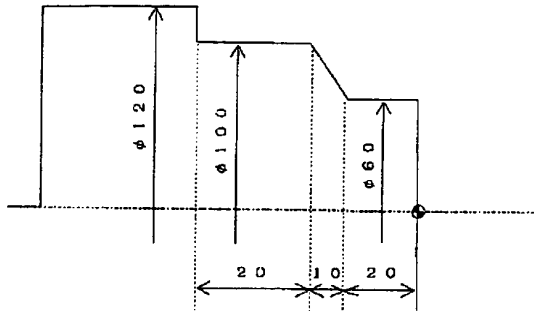


【図35】

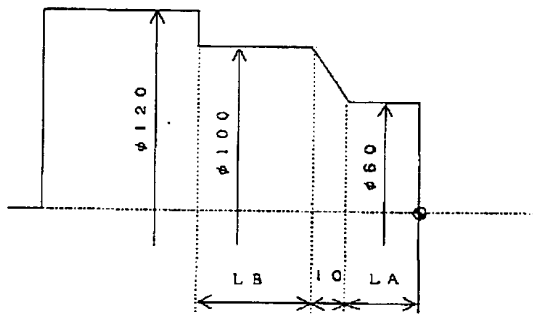
		t17					
		@D1	@D2				
t18 ~	TP1	60	5				
t18 ~	TP2	70	5				
t18 ~	TP3	80	5				
t18 ~	TP4	100	7				
t18 ~	TP5	120	9				

【図23】

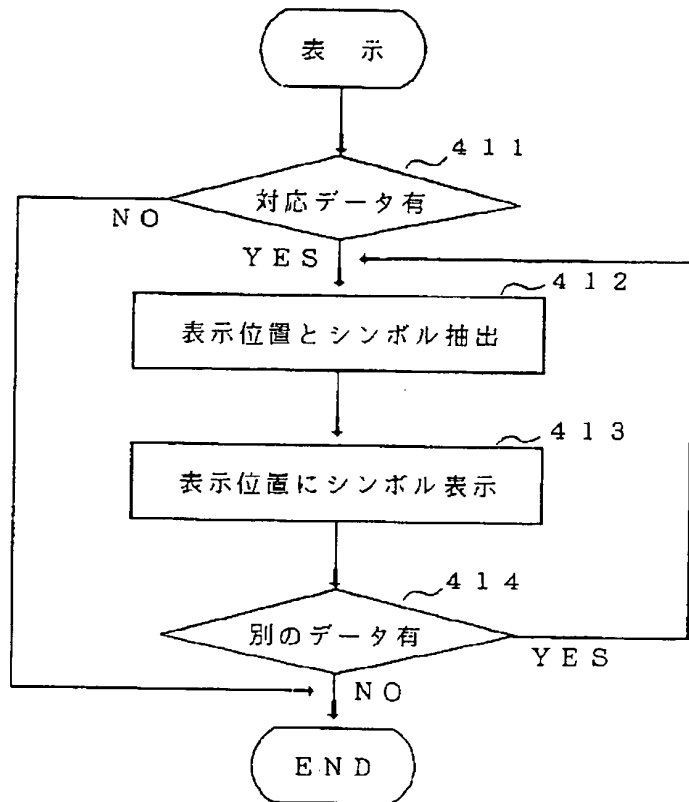
(1)



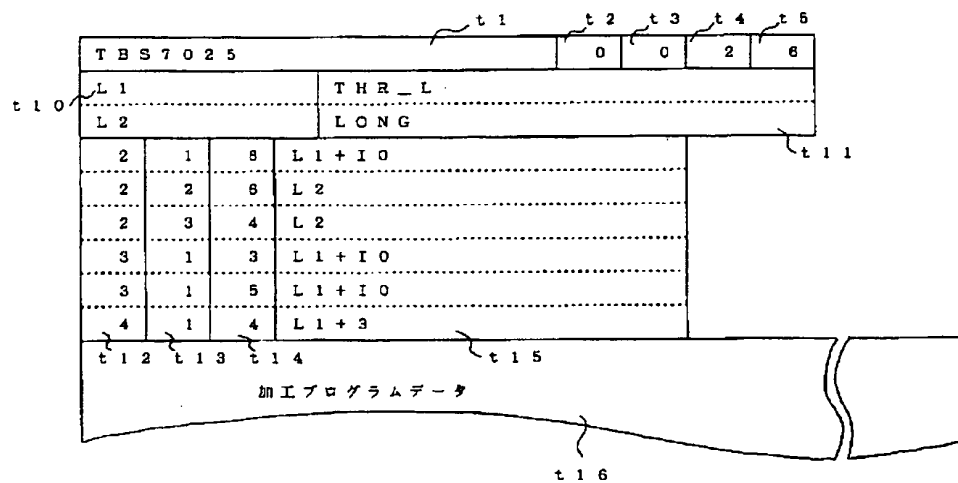
(2)



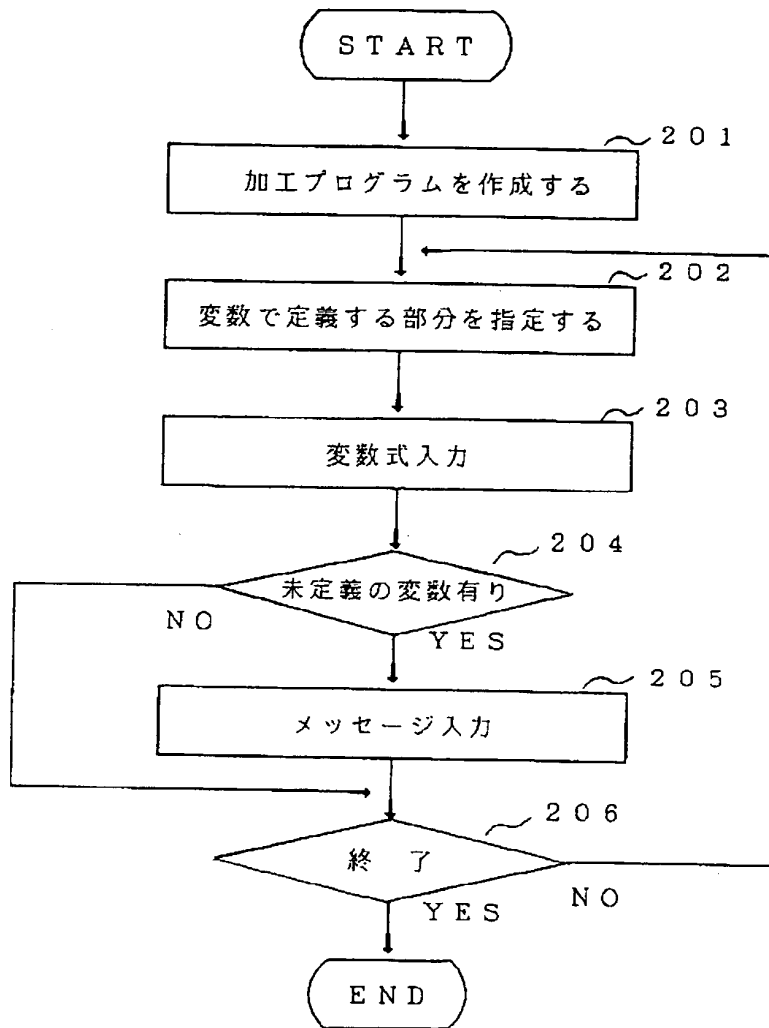
【図47】



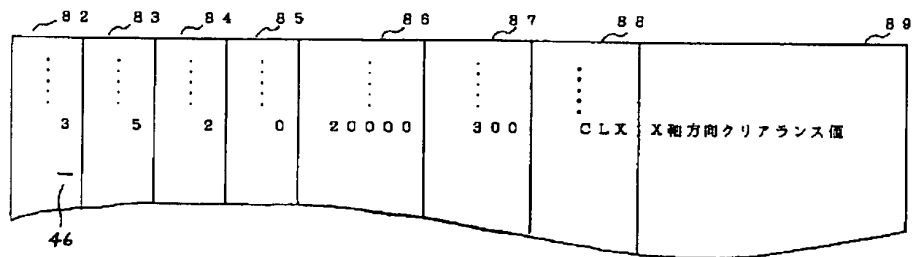
【図32】



【図24】



【図44】

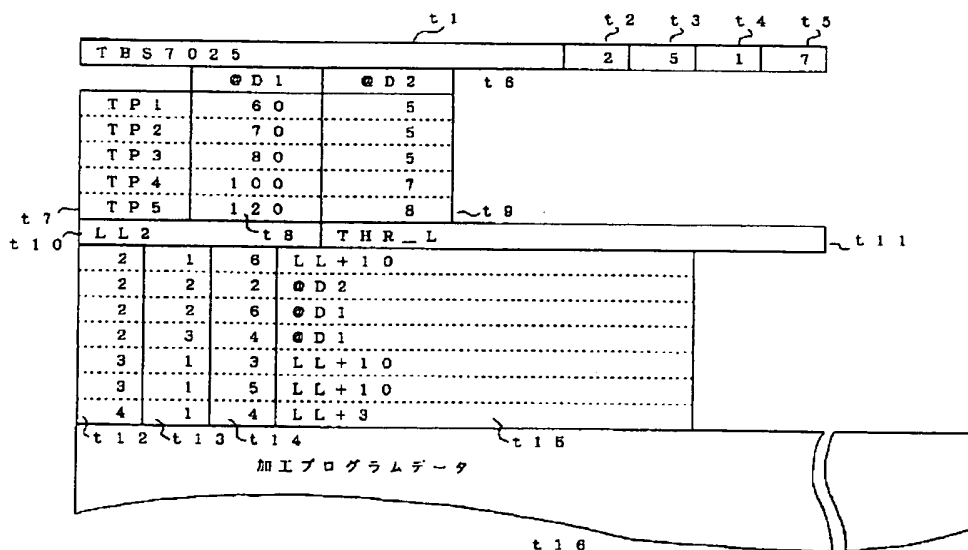


【図25】

PNo.	材質	外径	内径	素材長さ	回転数	仕上X	仕上Z	素材端面				
0	S45C	120	0.	90.	2000	0.2	0.1	0.				
PNo.	モード	#	切込X	切込Z		R周速	F周速	R送り	R切込	R工具	F工具	
1	BAR OUT	0	120.	0.		130	200	0.3	2.5	1	2	
SEQ	形	前コーナ	始点X	始点Z	終点X	終点Z	後コーナ/\$	半径R/角度	粗さ			
1	LIN		◆	◆	60.	20.	~ 4 6	◆	▼▼4			
2	TPR		60.	20.	100.	30.			▼▼4			
3	LIN		◆	◆	100.	50.		◆	▼▼4			
PNo.	モード	パーツカウント	復帰	ワークNo.	連続	回数	シフト量					
2	END	0	0		0	0	0.					

登 録		印 字	グループ化	変数定義
-----	--	-----	-------	------

【図36】



【図26】

PNo.	材質	外径	内径	素材長さ	回転数	仕上X	仕上Z	素材端面			
0	S45C	120	0.	90.	2000	0.2	0.1	0.			
PNo.	モード	#	切込X	切込Z		R周速	F周速	R送り	R切込	R工具	F工具
1	BAR OUT	0	120.	0.		130	200	0.3	2.5	1	2
SEQ	形	前コーナ	始点X	始点Z	終点X	終点Z	後コーナ/\$	半径R/角度	粗さ		
1	LIN		◆	◆	60.	20. ~ d1		◆	▼▼4		
2	TPR		60.	20. ~ d2	100.	30. ~ d3			▼▼4		
3	LIN		◆	◆	100.	50. ~ d4		◆	▼▼4		
PNo.	モード	パーツカウント	復帰	ワークNo.	連続	回数	シフト量				
2	END	0	0		0	0	0.				

【図29】

SEQ	形	前コーナ	始点X	始点Z	終点X	終点Z	後コーナ/\$	半径R/角度	粗さ
1	LIN		◆	◆	60.	(1)		◆	▼▼4
2	TPR		60.	(2)	100.	(3)			▼▼4
3	LIN		◆	◆	100.	(4)		◆	▼▼4

<使用変数>

LA: STEP_1L

LB: STEP_2L

<変数定数部>

(1): LA

(2): LA

(3): LA+10

(4): LA+LB+10

【図31】

PNo.	材質	外径	内径	素材長さ	回転数	仕上X	仕上Z	素材端面				
0	S45C	80.	0.	155.	2000	0.2	0.1	5.				
PNo.	モード				R周速	F周速	R送り	R切込	R工具	F工具		
1	EDG FCE				110	200	0.3	2.	1	2		
SEQ				始点X	始点Z	終点X	終点Z	粗さ				
1				80.	5.	0.	0.	▼▼3				
PNo.	モード	#	切込X	切込Z			R周速	F周速	R送り	R切込	R工具	F工具
2	BA2 OUT	0	80.	0.			130	200	0.3	2.5	3	4
SEQ	形	前コーナ	始点X	始点Z	終点X	終点Z	後コーナ/S		半径R/角度	粗さ		
1	LIN	C5.	◆	◆	50.	(1)			◆	▼▼3		
2	LIN	C5.	◆	◆	70.	(2)			◆	▼▼3		
3	凹		70.	(3)	80.	125.			50.	▼▼3		
PNo.	モード	#	数	ピッチ	溝幅	仕上代	R周速	F周速	送り	切込	R工具	F工具
3	GRV OUT	0	1	0	10	◆	◆	120	0.08	2.	◆	5
SEQ	前コーナ		始点X	始点Z	終点X	終点Z	後コーナ		角度	粗さ		
1			50.	(4)	40.	(5)						
PNo.	モード	#	チャンファ	リード	角度	条数	高さ	回数	周速	切込	工具	
4	THR OUT	0	0	2.	60	1	1.299	10	120	0.3	6	
SEQ				始点X	始点Z	終点X	終点Z					
1				50.	0.	50.	(6)					
PNo.	モード	パーツカウント		復帰	ワークNo.	連続	回数	シフト量				
5	END	0		0	0	0	0	0				

(1) L1+10

(2) L2

(3) L2

(4) L1+10

(5) L1+10

(6) L1+3

【図55】

ASCII コード (大文字 Alphabet)

A	B	C	D	E	F	G
0x41	0x42	0x43	0x44	0x45	0x46	0x47
H	I	J	K	L	M	N
0x48	0x49	0x4A	0x4B	0x4C	0x4D	0x4E
O	P	Q	R	S	T	U
0x4F	0x50	0x51	0x52	0x53	0x54	0x55
V	W	X	Y	Z	EOB	
0x58	0x59	0x5A	0x5B	0x5C	0x5D	

(注) EOB: End of Block

【図34】

PNo.	材質	外径	内径	素材長さ	回転数	仕上X	仕上Z	素材端面				
0	S45C	80.	0.	155.	2000	0.2	0.1	5.				
PNo.	モード				R周速	F周速	R送り	R切込	R工具	F工具		
1	EDG FCE				110	200	0.3	2.	1	2		
SEQ			始点X	始点Z	終点X	終点Z				粗さ		
1			80.	5.	0.	0.				▼▼3		
PNo.	モード	#	切込X	切込Z			R周速	F周速	R送り	R切込	R工具	F工具
2	BAR OUT	0	80.	0.			130	200	0.3	2.5	3	4
SEQ	形	前コーナ	始点X	始点Z	終点X	終点Z	後コーナ/S	半径R/角度	粗さ			
1	LIN	C5.	◆	◆	50.	(1)		◆	▼▼3			
2	LIN	(2)	◆	◆	70.	(3)		◆	▼▼3			
3	凹		70.	(4)	80.	125.		50.	▼▼3			
PNo.	モード	#	数	ピッチ	溝幅	仕上代	R周速	F周速	送り	切込	R工具	F工具
3	GRV OUT	0	1	0	10	◆	◆	120	0.08	2.	◆	5
SEQ			前コーナ	始点X	始点Z	終点X	終点Z	後コーナ	角度	粗さ		
1				50.	(5)	40.	(6)					
PNo.	モード	#	チャンファ	リード	角度	条数	高さ	回数	周速	切込	工具	
4	THR OUT	0	0	2.	60	1	1.299	10	120	0.3	6	
SEQ			始点X	始点Z	終点X	終点Z						
1			50.	0.	50.	(7)						
PNo.	モード	パーツカウント		復帰	ワークNo.	連続	回数	シフト量				
5	END	0		0	0	0	0	0				

(1) LL+10

(2) @D2

(3) @D1

(4) @D1

(5) LL+10

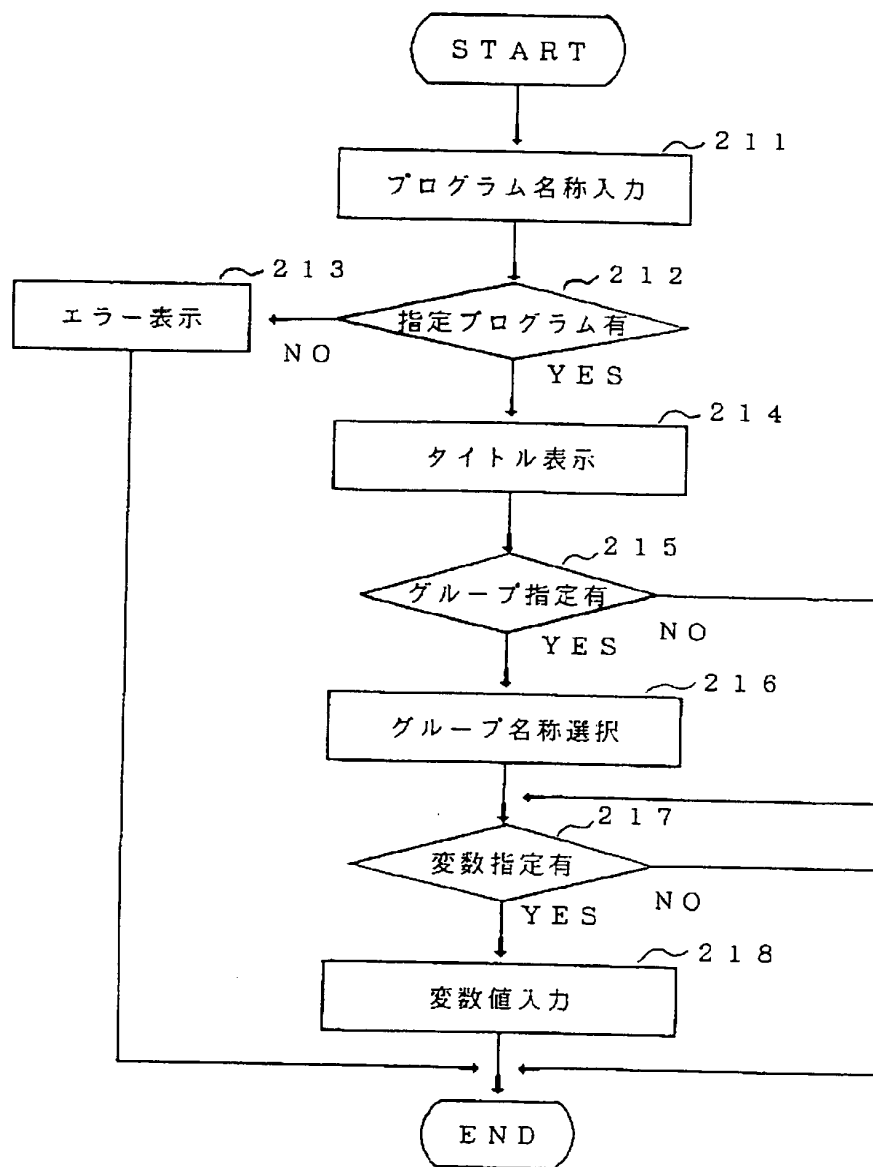
(6) LL+10

(7) LL+3

【図37】

PNo.	材質	外径	内径	素材長さ	回転数	仕上X	仕上Z	素材端面
0								

【図38】



【図39】

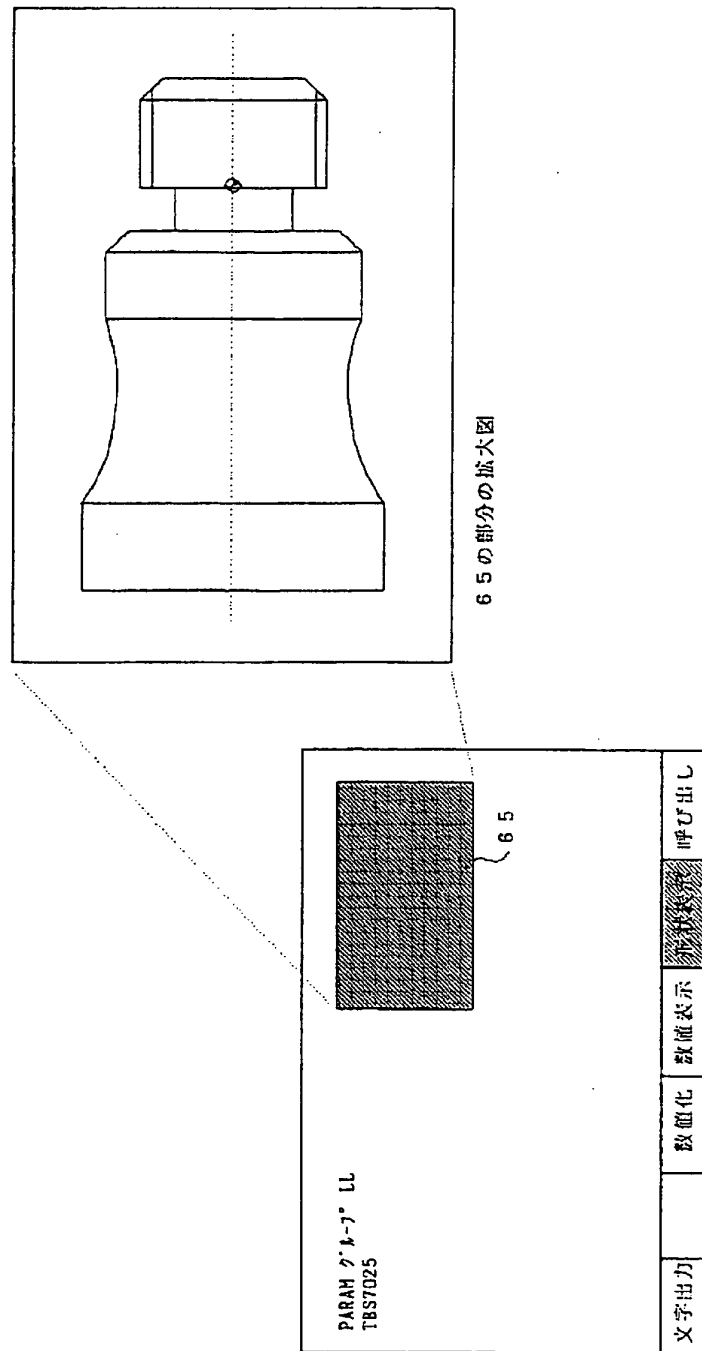
PARAM グループ LL					
TBS7025 TP3 50.					
文字出力		数値化	数値表示	形状表示	呼び出し

【図45】

— (1, 1) —	— (1, 2) —	— (1, 3) —
— (2, 1) —	— (2, 2) —	— (2, 3) —
— (3, 1) —	— (3, 2) —	— (3, 3) —
— (4, 1) —	— (4, 2) —	— (4, 3) —
— (5, 1) —	CLX 152Ω	— (5, 3) —
— (6, 1) —	— (6, 2) —	— (6, 3) —
— (7, 1) —	— (7, 2) —	— (7, 3) —
— (8, 1) —	— (8, 2) —	— (8, 3) —

X軸方向クリアランス値 ⁷⁶
 許容範囲 0~20000 標準値 ⁷⁸ 300
₇₇

【図40】



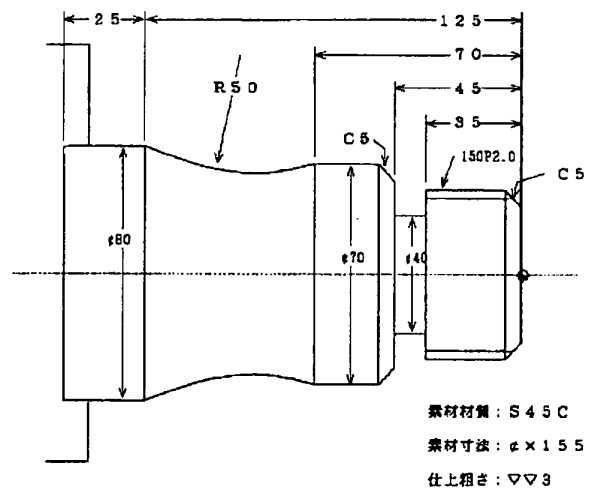
【図41】

PNo.	材質	外径	内径	素材長さ	回転数	仕上X	仕上Z	素材端面				
0	S45C	80.	0.	155.	2000	0.2	0.1	5.				
PNo.	モード				R周速	F周速	R送り	R切込	R工具	F工具		
1	EDG FCE				110	200	0.3	2.	1	2		
SEQ			始点X	始点Z	終点X	終点Z					粗さ	
1			80.	5.	0.	0.					▼▼3	
PNo.	モード	#	切込X	切込Z			R周速	F周速	R送り	R切込	R工具	F工具
2	BAR OUT	0	80.	0.			130	200	0.3	2.5	3	4
SEQ	形	前コーナ	始点X	始点Z	終点X	終点Z	後コーナ/\$	半径R/角度	粗さ			
1	LIN	C5.	◆	◆	50.	60		◆	▼▼3			
2	LIN	C5	◆	◆	70.	80		◆	▼▼3			
3	凹		70.	80	80.	128		50.	▼▼3			
PNo.	モード	#	数	ピッチ	溝幅	仕上代	R周速	F周速	送り	切込	R工具	F工具
3	GRV OUT	0	1	0	10	◆	◆	120	0.08	2.	◆	5
SEQ			前コーナ	始点X	始点Z	終点X	終点Z	後コーナ	角度	粗さ		
1				50.	60	40.	80					
PNo.	モード	#	チャンファ	リード	角度	条数	高さ	回数	周速	切込	工具	
4	THR OUT	0	0	2.	60	1	1.299	10	120	0.3	6	
SEQ			始点X	始点Z	終点X	終点Z						
1			50.	0.	50.	53						
PNo.	モード	パーツカウント		復帰	ワークNo.	連続	回数	シフト量				
5	END	0		0	0	0	0	0				

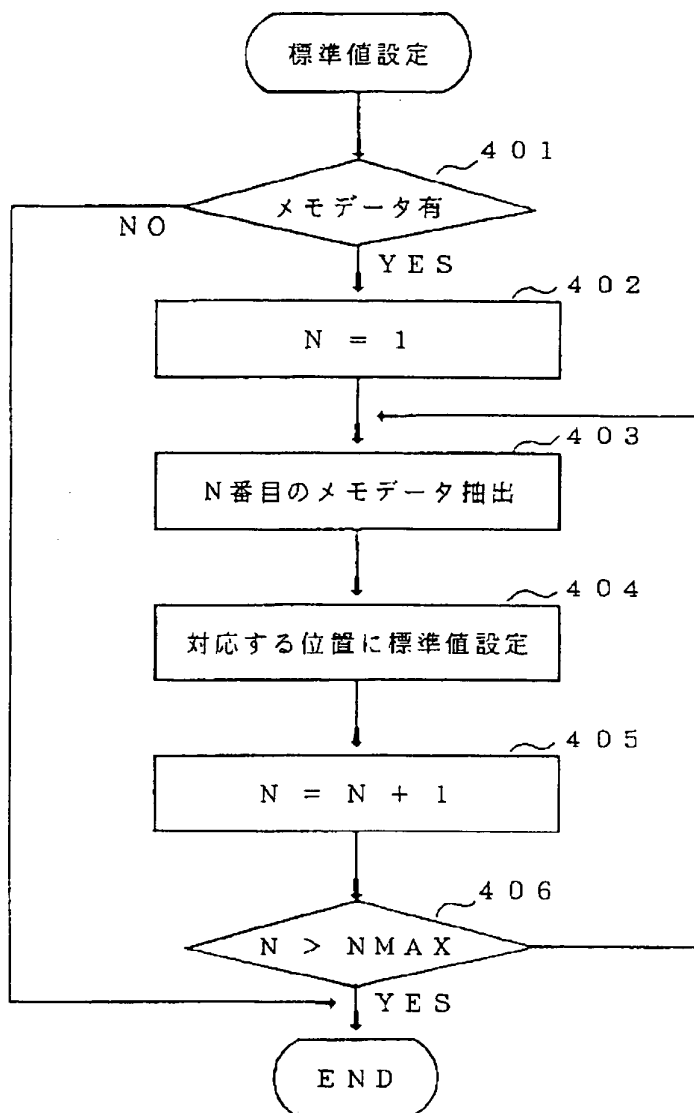
【図66】

[PROGRAM FILE]		IN/OUT 5. 1/1	
PROGRAM ENTRY	8	REMAIN	32
CHARACTER	1500	REMAIN	16000
<PROGRAM>	<CHR>	<ST>	<COMMENT>
90	123		TEST CUT NO.12
100			
200			
300			
400			
500			
0(<input type="checkbox"/>) COMMENT()			
INPUT	OUTPUT	ERASE	COPY
			FILE

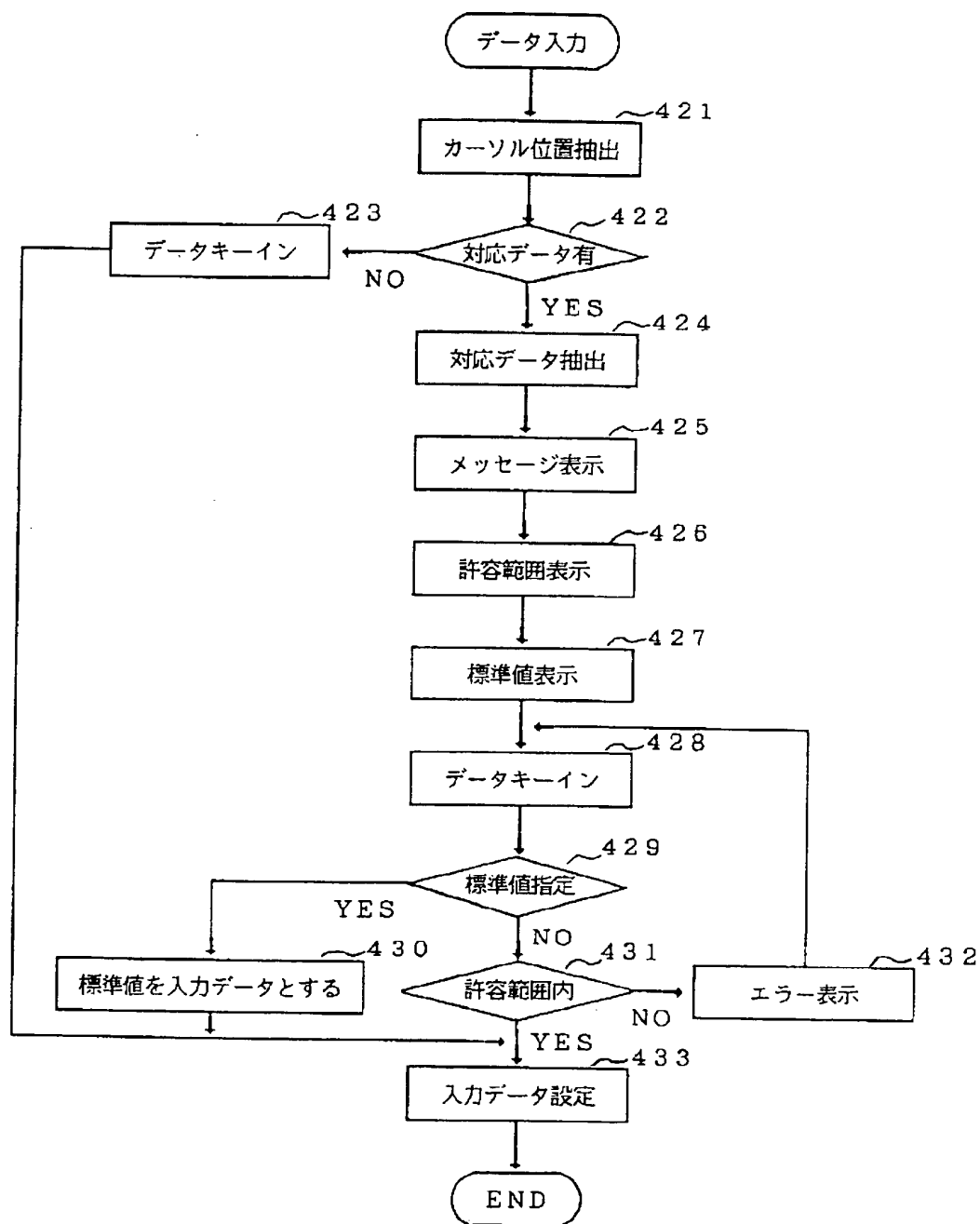
【図68】



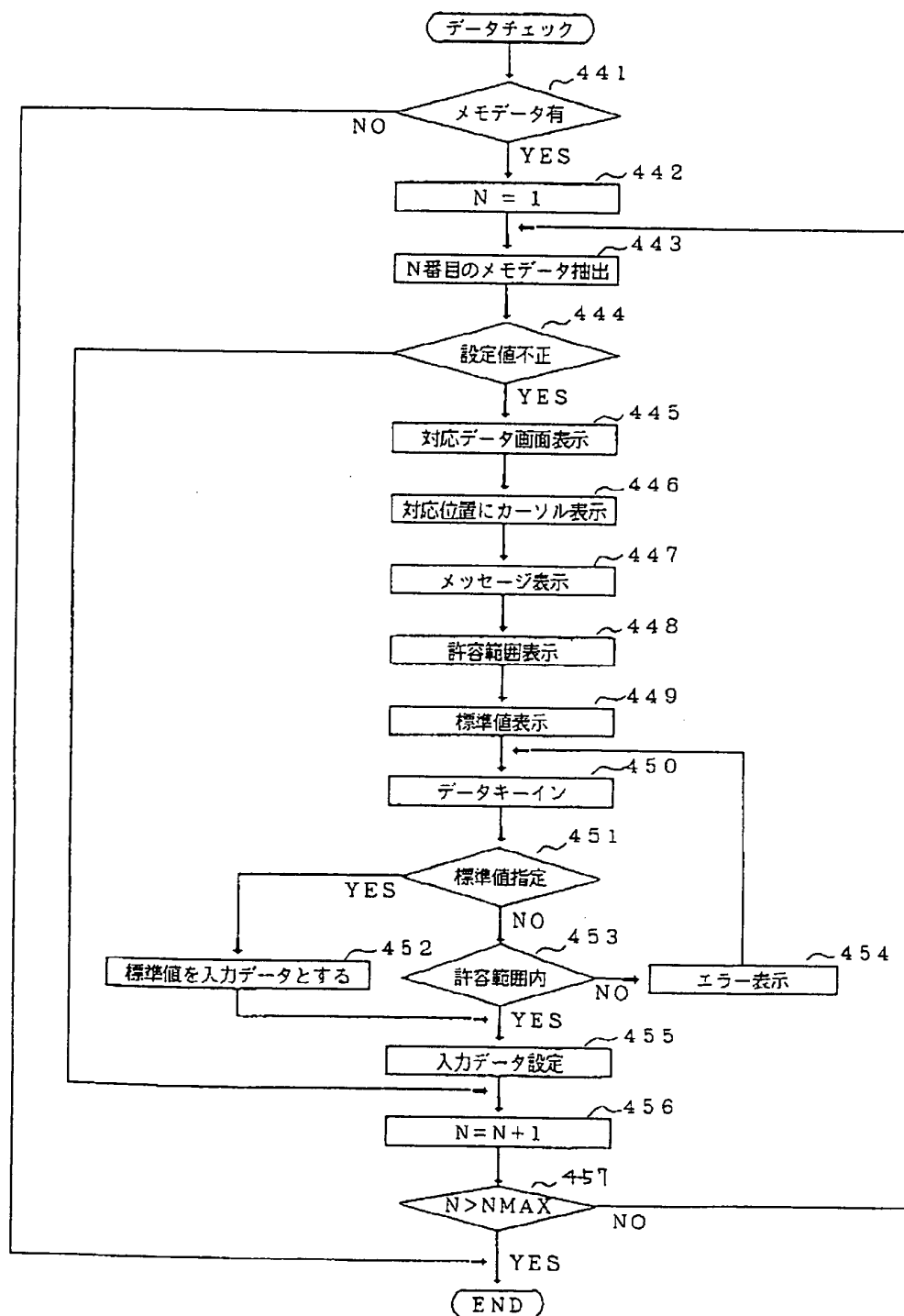
【図46】



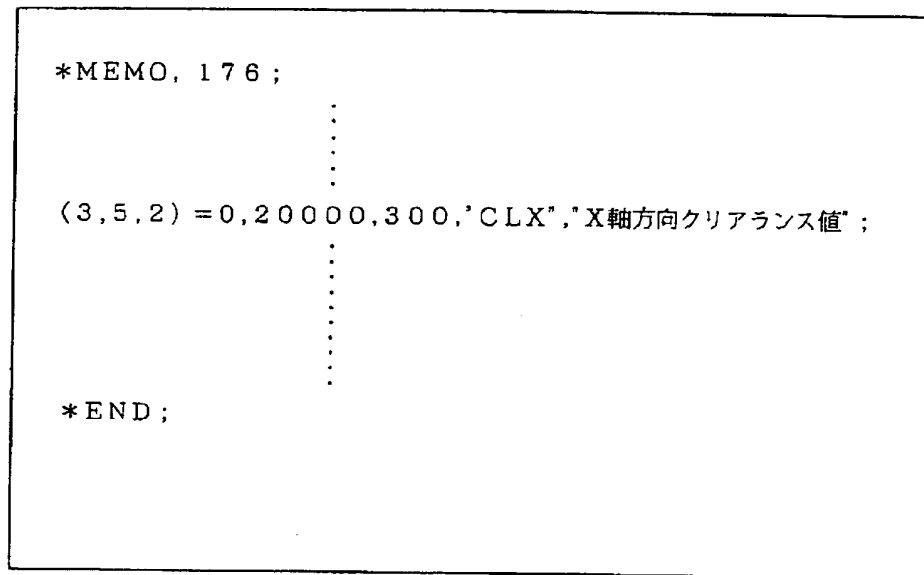
【図48】



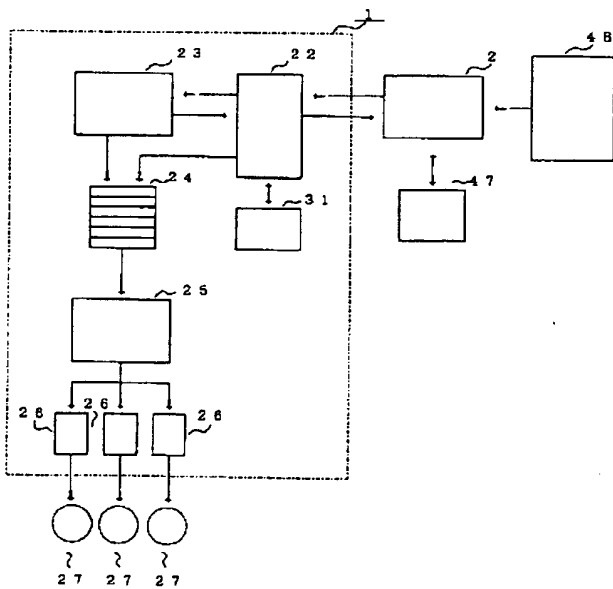
【図49】



【図50】



【図51】



【図57】

(A)

G01 X30400 Z-29600 F5711;

(B)

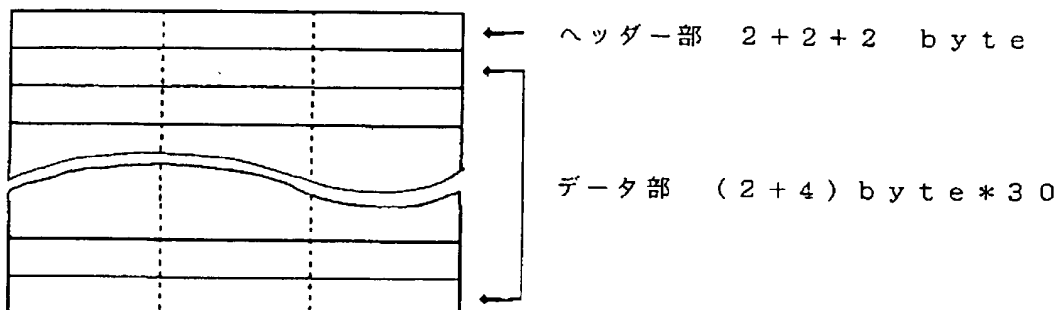
8004	00080001	← ヘッダー部
0047	00000001	G01
005A	FFFF8C60	Z-29600
0058	000076C0	X30400
0048	00001A37	F5711
000A	00000000	EOB

(C)

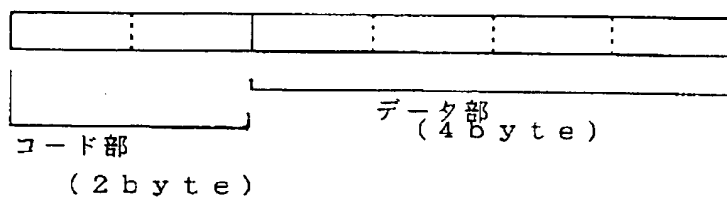
800400080001
004700000001
0050FFFF8C60
0058000076C0
004800001A37
000A0000000048;

【図52】

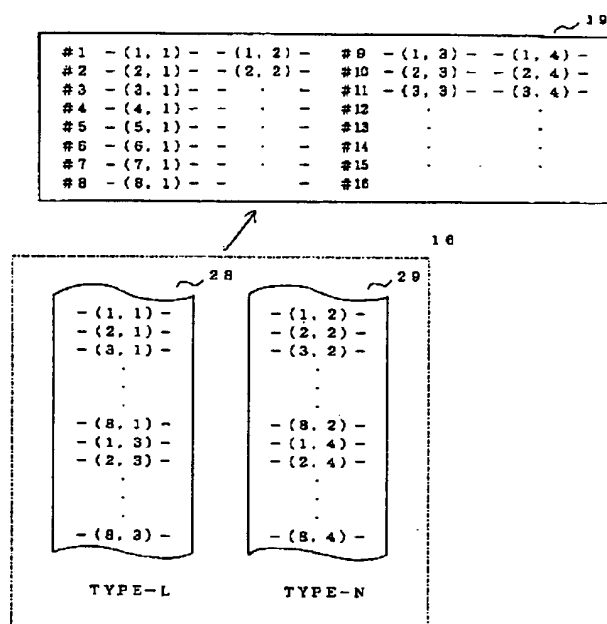
(A)



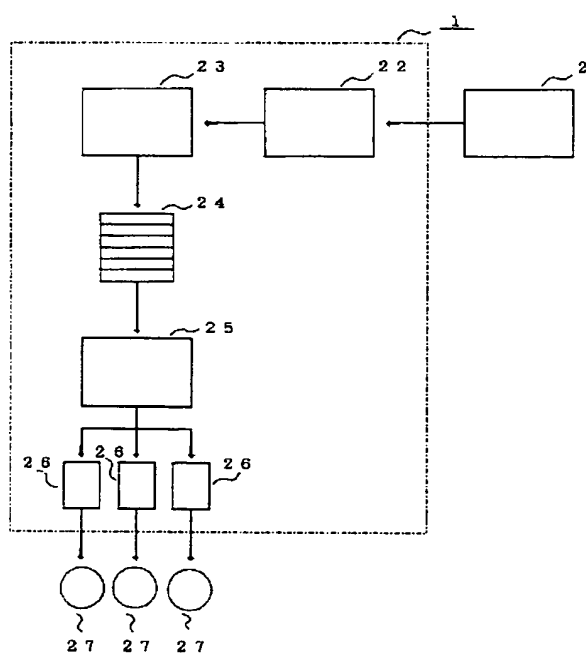
(B)



【図67】

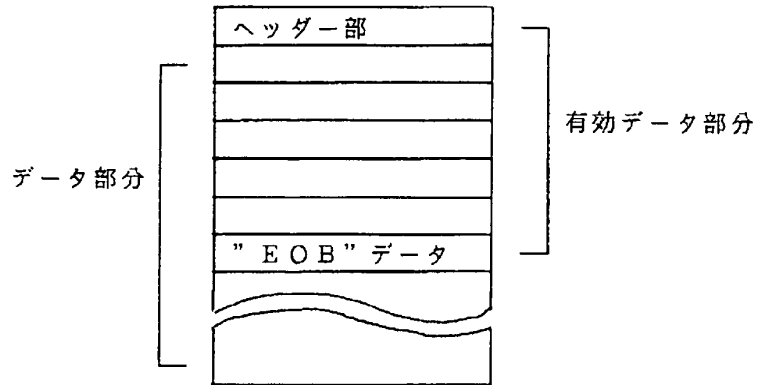


【図70】



【図53】

(A)



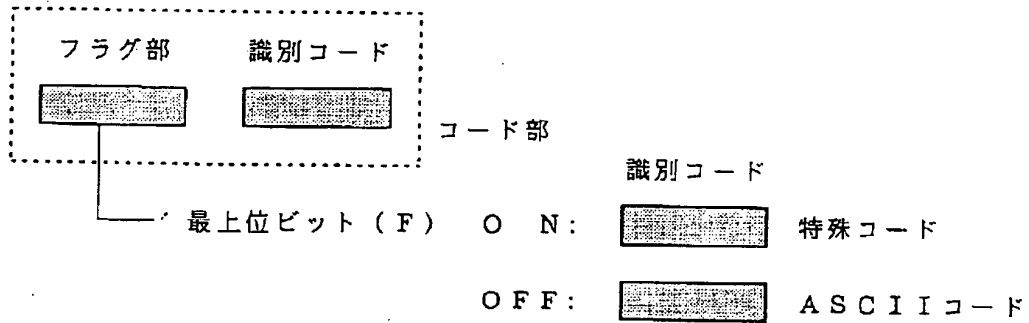
(B)

" E O B " データ

0x00	0x0A	0x00	0x00	0x00	0x00
------	------	------	------	------	------

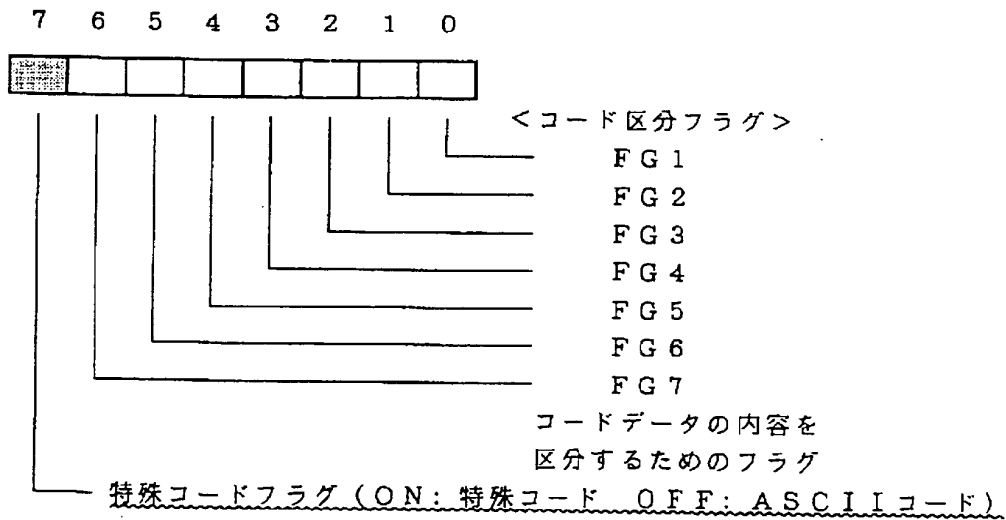
【図54】

(A)



(B)

(1) フラグ部

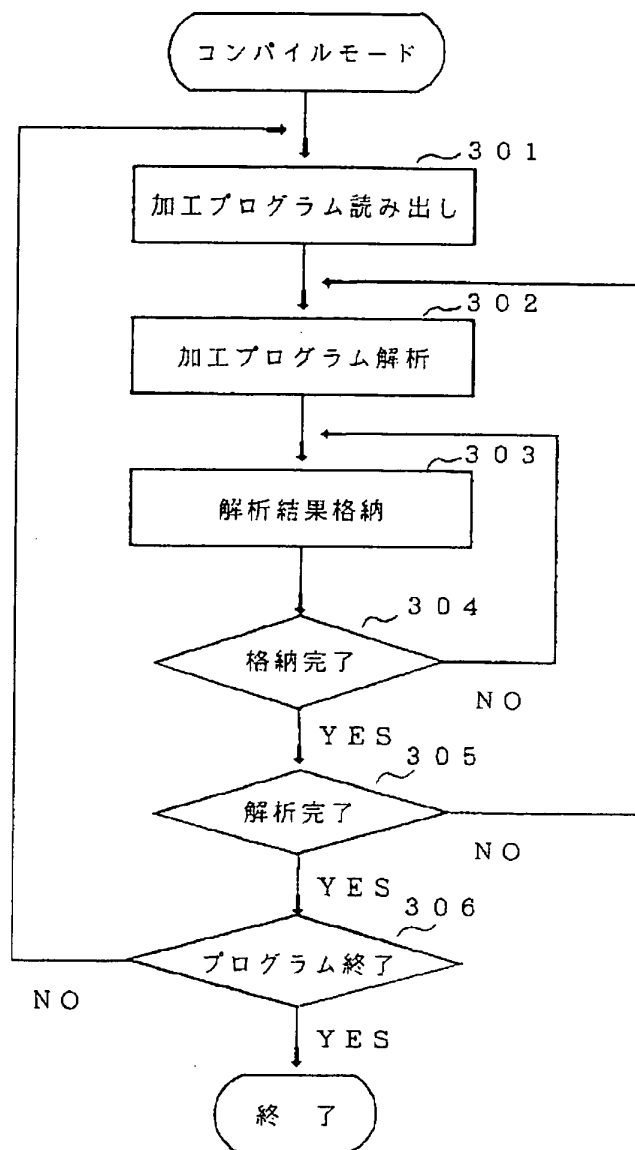


【図56】

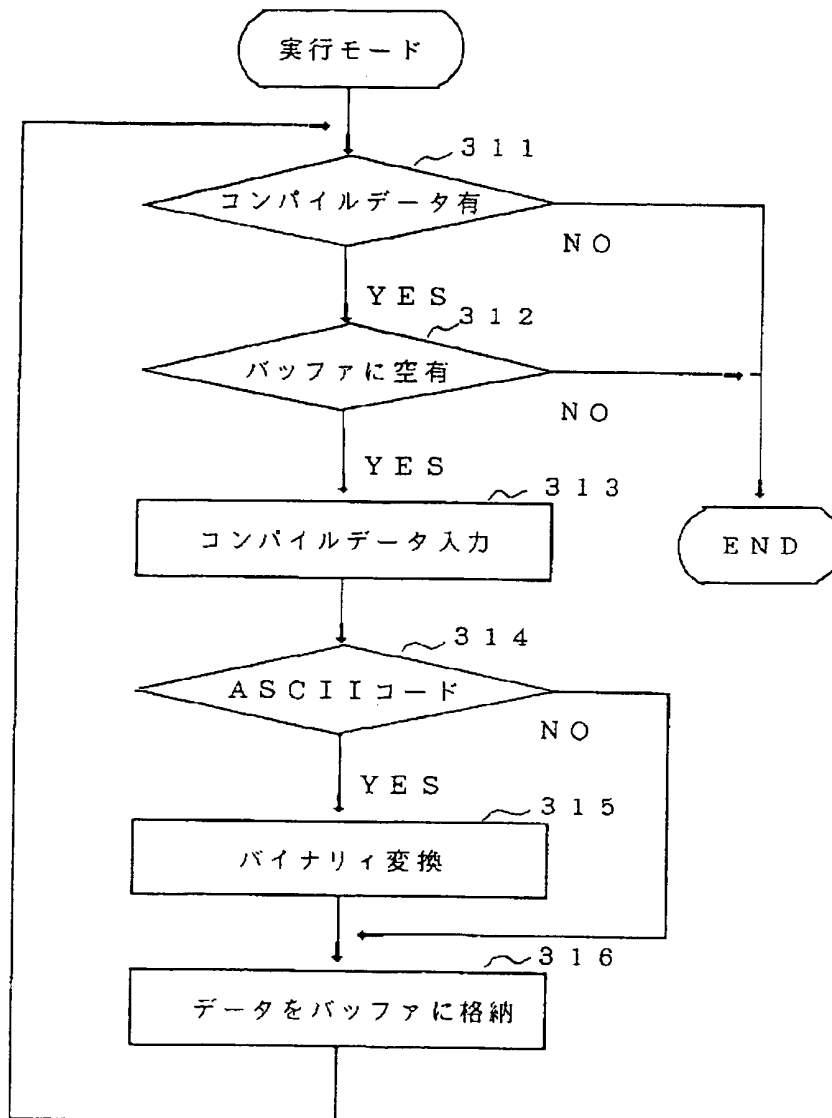
コード	内 容
01	ワークNo.
02	プロセスNo. + 荒/仕上加工区分
03	加工モード+加工部
04	待機用プログラムNo. (待機する際に実行するプログラムNo.)
05	現工具アドレス (この工具が現在装着されているはず)
06	使用工具アドレス (本プロセスで使用する工具のアドレス)
07	次回使用工具アドレス (次に使用する予定の工具アドレス)
08	ファイルシステム情報 (シークポインター+キャラクターポインター)
09	次プロセス使用工具アドレス (次プロセスで使用する工具)
0A	ASCII EOBコード (有効データの最後であることを示す)
0B	次プロセスの プロセスNo. + 荒/仕上加工区分
0C	次プロセスの 加工モード+加工部
10	円弧長 (G02, G03 指令時の円弧の長さ)
11	チャンファ量 (G32 指令時のネジ切りのチャンファリング距離)
12	チャンファ角度 (G32 指令時のネジ切りのチャンファリング角度)
13	糸数オフセット角度 (G32 指令時のネジ切りの糸数オフセット角度)
14	ノーズR補正值 (Z軸方向の値)
15	ノーズR補正值 (X軸方向の値)
16	タレットのポケットNo. を出力する
17	工具の補正No. を出力する
18	補正 (ノーズR補正以外) Z軸方向 ON/OFF
19	補正 (ノーズR補正以外) X軸方向 ON/OFF
1C	相手側HEADの ノーズR補正值 (Z軸方向の値)
1D	相手側HEADの ノーズR補正值 (X軸方向の値)
1E	相手側HEADの タレットのポケットNo.
20	ブロック制御データ (各ブロックの制御用フラグ)
21	計測制御コード
22	Gモーダルデータ (EIA専用・表示用)
28	相手側HEADの補正 (ノーズR補正以外) Z軸方向 ON/OFF
29	相手側HEADの補正 (ノーズR補正以外) X軸方向 ON/OFF
30	素材最大外径値 (自動プログラムの共通データの設定値)
31	素材最小内径値 (自動プログラムの共通データの設定値)
32	主軸クランプ回転数上限値 (自動プログラムの共通データの設定値)
33	主軸クランプ回転数上限値 (EIA専用)
41~5Aまでは、ASCIIコード用 (大文字) に リザーブ (特殊コードには使用不可)	

コードの値は16進数

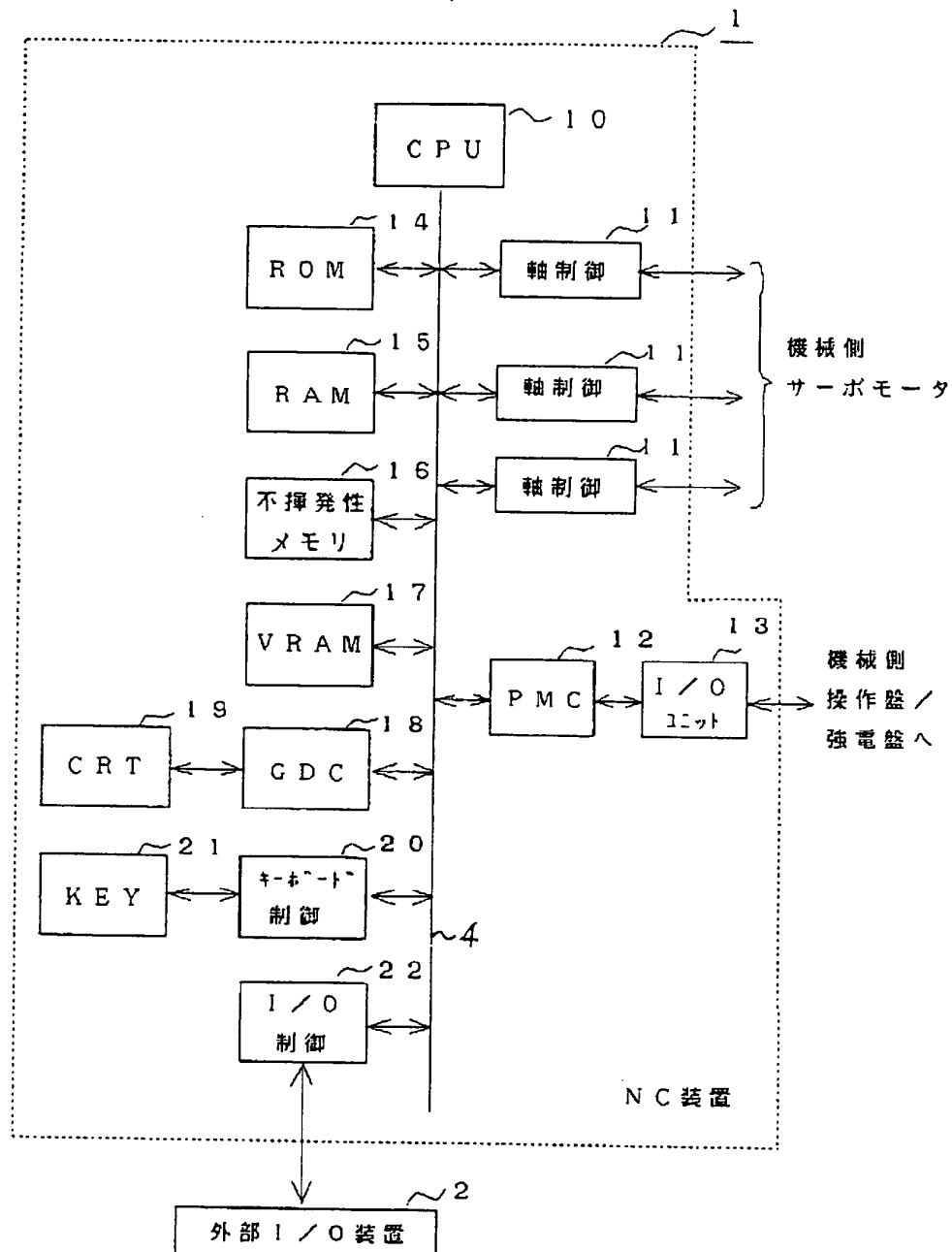
【図58】



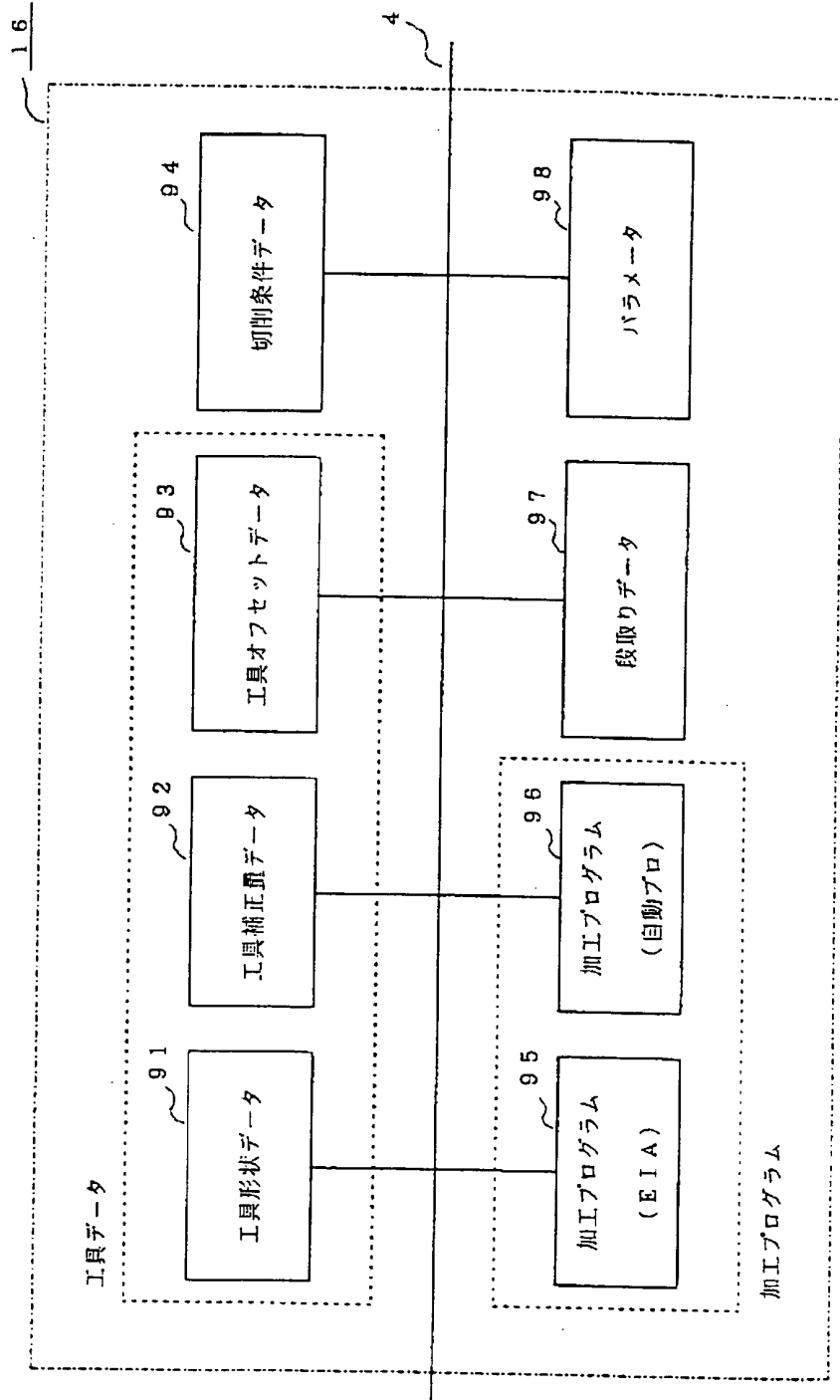
【図59】



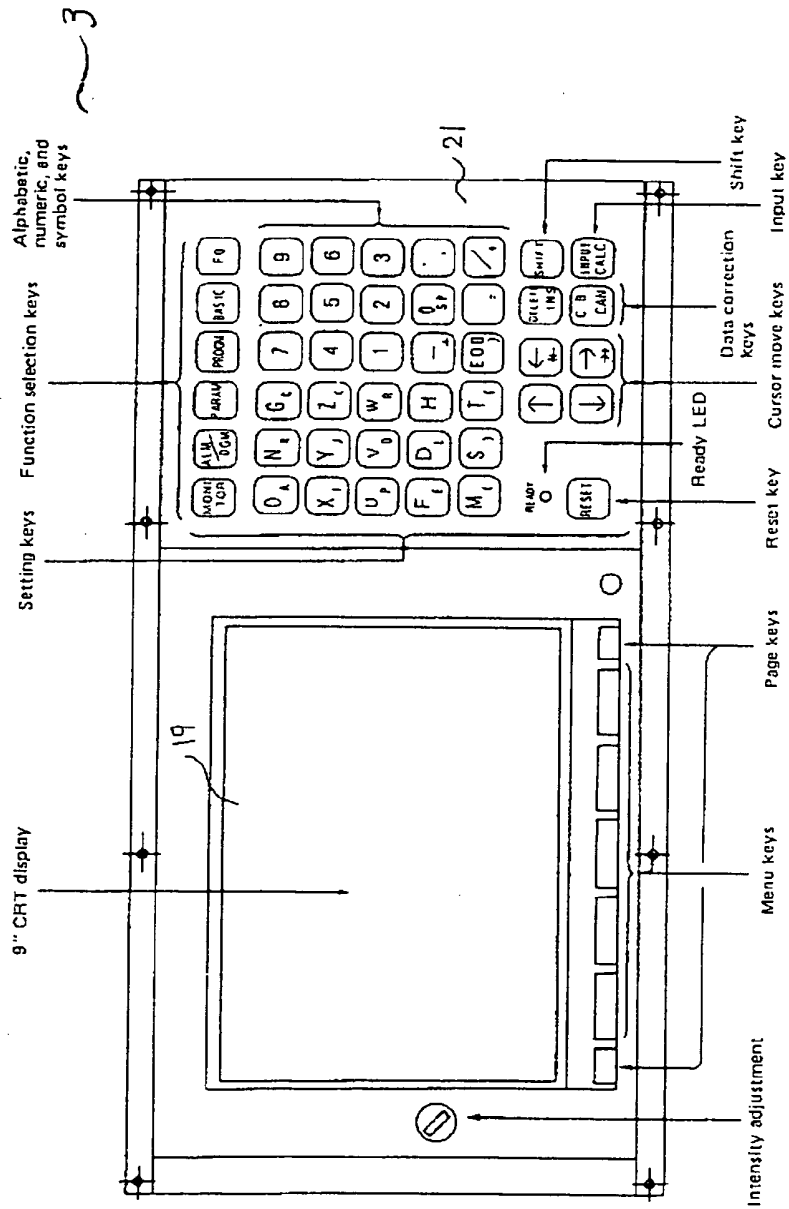
【図60】



【図61】



【図 6 2】



【図69】

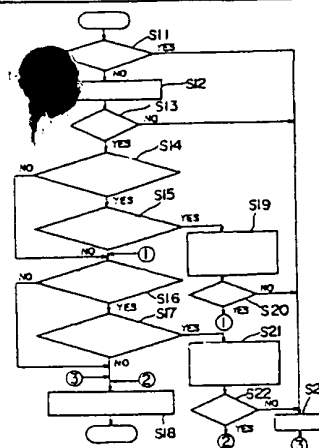
PNo.	材質	外径	内径	素材長さ	回転数	仕上X	仕上Z	素材端面				
0	S45C	80.	0.	155.	2000	0.2	0.1	5.				
PNo.	モード					R周速	F周速	R送り	R切込	R工具	F工具	
1	EDG FCE					110	200	0.3	2.	1	2	
SEQ					始点X	始点Z	終点X	終点Z	粗さ			
1					80.	5.	0.	0.	▼▼3			
PNo.	モード	#	切込X	切込Z			R周速	F周速	R送り	R切込	R工具	F工具
2	BAR OUT	0	80.	0.			130	200	0.3	2.5	3	4
SEQ	形	前コーナ	始点X	始点Z	終点X	終点Z	後コーナ/\$	半径R/角度	粗さ			
1	LIN	C5.	◆	◆	50.	45.		◆	▼▼3			
2	LIN	C5.	◆	◆	70.	70.		◆	▼▼3			
3	凹		70.	70.	80.	125.		50.	▼▼3			
PNo.	モード	#	数	ピッチ	溝幅	仕上代	R周速	F周速	送り	切込	R工具	F工具
3	GRV OUT	0	1	0	10	◆	◆	120	0.08	2.	◆	5
SEQ	前コーナ		始点X	始点Z	終点X	終点Z	後コーナ	角度	粗さ			
1			50.	45.	40.	45.						
PNo.	モード	#	チャンファ	リード	角度	条数	高さ	回数	周速	切込	工具	
4	THR OUT	0	0	2.	60	1	1.299	10	120	0.3	6	
SEQ			始点X	始点Z	終点X	終点Z						
1			50.	0.	50.	38.						
PNo.	モード	パーツカウント	復帰	ワークNo.	連続	回数	シフト量					
5	END	0	0	0	0	0	0					

(54) METHOD FOR GENERATING INTERFERENCE EVASION ROUTE

(11) 5-134732 (A) (43) 1.6.1993 (19) JP
 (21) Appl. No. 3-295573 (22) 12.11.1991
 (71) FANUC LTD (72) TATSUYA OMI
 (51) Int. Cl⁵. G05B19/403, B25J9/10, B25J9/16

PURPOSE: To automatically generate an interference evasion route in short time in the operation program of a robot.

CONSTITUTION: When the robot interferes with another object between the initial and last positions of the robot, a temporary evasion point is obtained between the immediately preceding position and the immediately succeeding position of an interference area (S12). When the temporary evasion point interferes with the initial position (S14), it is discriminated whether the immediately preceding position interferes with the temporary evasion point or not (S15), the immediately preceding position is regarded as the initial position when interference exist, the temporary evasion point is re-obtained since the temporary evasion point is regarded as the last position and a case without interference is retrieved (S19). The route connecting the initial position with the temporary evasion point is established and, then, the route connecting the temporary evasion point with the last position is established in a same way (S16, S17 and S21). Thus, the interference evasion route is generated as the route connecting the initial position, the evasion point and the last position.



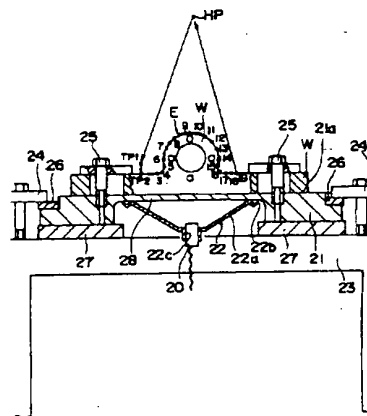
A: start. B: end. S11: more than designated times? S12: to obtain temporary evasion point. S13, S20 and S22: success? S14: Does interference exist between the last position and temporary evasion point? S15: Does interference exist between immediately preceding position and the temporary evasion point? S16: to set evasion point and to write a program. S17: S11-S15 are executed with initial position and the last position as immediately preceding position and temporary evasion point. S19: S11-S17 are executed with initial position and the last position as temporary evasion point and immediately succeeding position.

(54) WORKING ROBOT

(11) 5-134733 (A) (43) 1.6.1993 (19) JP
 (21) Appl. No. 3-298635 (22) 14.11.1991
 (71) HITACHI CONSTR MACH CO LTD (72) KUNIHICO YOSHIDA(1)
 (51) Int. Cl⁵. G05B19/403, B24B27/00, B25J9/10, B25J13/08, B25J19/02

PURPOSE: To provide a working robot executing uniformized deburring in work in spite of the size of a shape error in work.

CONSTITUTION: A sound sensor such as a microphone 20, etc., is attached in the back surface of a jig 21 where work W is fitted and work sound to work W by a working tool such as a grinder gripped by the working robot, etc., is detected. A work start position which is previously taught is compared with the position of the grinder when work sound is obtained and a work route which is designated at first is corrected by the difference between the two. A sound-collecting horn 22 is arranged in the periphery of the sound sensor and a thin resonant disk part 28 is provided at the jig 21 so that the sensitivity for detecting work sound is enhanced.

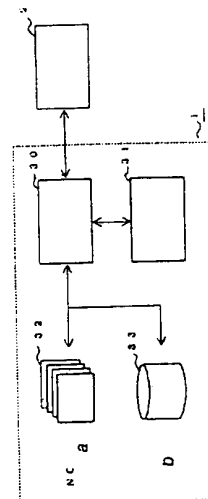


(54) NUMERICAL CONTROLLER AND CONTROL METHOD THEREFOR

(11) 5-134734 (A) (43) 1.6.1993 (19) JP
 (21) Appl. No. 4-62143 (22) 18.3.1992 (33) JP (31) 91u.74367 (32) 17.9.1991
 (71) MITSUBISHI ELECTRIC CORP (72) TOMOMITSU NIWA
 (51) Int. Cl⁵. G05B19/403, G05B19/405

PURPOSE: To enable input/output to be executed concerning all kinds of data in an NC controller between the NC controller and an external input/output device and to execute the input/output by means of a character code by storing the respective kinds of data inside the NC controller as two-dimensional or one-dimensional array data in accordance with respective screens so as to convert the data in a character code conversion processing part.

CONSTITUTION: A data input/output control part 30 is a part controlling the input/output of data between the NC controller and the external input/output device 2 and is connected to the character code conversion processing part 31 so that data except the character code in the NC controller 1 is converted into the character code and the character code which is inputted from the external input/output device is converted into data in the NC controller 1. The internal data of an NC data storing area 32 is stored by a prescribed format as array data corresponding to the respective screens of the NC controller 1. The respective kinds of screen data is constituted of a header information part consisting of screen numbers and also a data part consisting of the number of rows and columns in array, etc.



(a): NC internal data. (b): file data